Machine Translation for Professional Translators

Thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zurich
for the degree of Doctor of Philosophy

by
Samuel Läubli

Accepted in the fall semester 2020
on the recommendation of the doctoral committee composed of
Prof. Dr. Martin Volk (main supervisor)
Prof. Dr. Rico Sennrich
Prof. Dr. Joss Moorkens

Zurich, 2020

Abstract

Professional translation was one of the first computational settings in which human users were paired with intelligent machine agents. Almost seven decades after the first machine translation (MT) experiments in the early 1950s, MT has made astounding progress due to advances in neural modelling, but still faces scepticism and slow adoption among professional translators. In this thesis, we investigate three key problems that previous work has found to impede the acceptance and efficient use of MT by professional translators: quality, presentation, and adaptability.

The main contributions of this thesis are:

- A re-assessment of Hassan et al.'s (2018) claim of human–machine parity in Chinese to English news translation. Our empirical analysis confirms parity in sentence-level adequacy, but shows that professional human translation is superior to machine translation in sentence-level fluency as well as document-level adequacy and fluency. Our findings have motivated a shift towards document-level evaluation in the News Translation task at the Conference on Machine Translation (WMT), a major venue for competitive MT research (Barrault et al., 2019).
- A controlled experiment on how the presentation of translation suggestions in translation user interfaces (UIs) affects speed and accuracy in 20 professional translators. We find that a top-and-bottom arrangement of source and target sentences is most time efficient for interleaved reading and writing, even if most subjects use a left-and-right arrangement in their daily work. Document-level UIs that do not split texts into sentences lead to the highest accuracy in revision for anaphoric relations between sentences.
- Infix Generation, a multi-encoder method to incorporate prior knowledge in the form of partial translations into neural MT. In contrast to prefix decoding (Knowles and Koehn, 2016), it allows the inclusion of a (possibly empty) suffix, and achieves a speedup of an order of magnitude over Grid Beam Search (Hokamp and Liu, 2017), a constrained decoding method with similar capacity.

Acknowledgements

If it wasn't for Martin Volk's brilliant course on MT and parallel corpora in the spring semester of 2009, I would have probably never discovered how fascinating – and how challenging, indeed – it is to create software that translates text from one natural language into another. I am indebted to Martin for being one of the best teachers I ever had, and for all of his support as the main supervisor of this thesis. Likewise, I owe deep gratitude to Rico Sennrich. Rico effectively rescued me from a motivational low at the very beginning of my PhD: I remember well how, after gaining hands-on experience with statistical MT in the software industry, I attended the 2016 MT Marathon in Prague, where I learned that neural methods had basically made the statistical paradigm – and much of my knowledge thereof – redundant. I knew I was in safe hands when Rico, who was largely responsible for this paradigm shift, offered to become my co-supervisor. I'm very grateful to Martin and Rico for letting me chase wild ideas in the intersection of MT and human–computer interaction, and for helping to turn these ideas into actual research experiments and results. I mean it when I say that I couldn't have wished for a better team of mentors.

Over the past years, I've also had the chance to learn about the human perspective on translation from many inspiring people. First and foremost, I would like to thank Maureen Ehrensberger-Dow and Gary Massey who, as early as in 2012, told me it would make little sense to look at isolated sentences when evaluating translations. It took the MT community six more years to arrive at this conclusion, and if the empirical findings presented in this thesis have somewhat contributed to that, this is also thanks to their motivation and interest in combining research on human translation with research on MT. The latter equally goes for Jean Nitzke and David Orrego-Carmona, to whom I'm grateful for invaluable discussions on translation process research and MT. I look forward to many more of those! To Joss Moorkens, I extend my gratitude for serving as the external examiner of this thesis, and for making helpful suggestions to improve it.

I would also like to thank the people I was fortunate to work with at Autodesk, namely Valéry Jacot and Patrice Ferrot, and at Lilt, notably Carmen Heger, Saša Hasan, Patrick Simianer, Joern Wuebker, and Spence Green. From the people I collaborated with in academic settings, a special thank you goes out to Mark Fishel.

Last but certainly most of all, I would like to thank my family and friends. Going through the ups and downs of a PhD will never even remotely compare to raising four kids on your own. To my mum, I don't know how you did it, but will be forever grateful.

Contents

List of Figures

List of Tables

Abbreviations

AIC	Akaike information criterion
BIC	Bayesian information criterion
BPE	Byte-pair Encoding (for MT, Sennrich et al., 2016b)
CAT	Computer-aided translation
CLT	Constrained LT (Susanto et al., 2020)
DA	Direct Assessment (Graham et al., 2013)
DBA	Dynamic Beam Allocation (Post and Vilar, 2018)
GBS	Grid Beam Search (Hokamp and Liu, 2017)
HCI	Human–computer interaction
HT	Human translation
IG	Infix Generation (Section 6.3)
IMT	Interactive MT
LSTM	Long Short-term Memory (Hochreiter and Schmidhuber, 1997)
LT	Levenshtein Transformer (Gu et al., 2019)
MT	Machine translation
NFR	Neural Fuzzy Repair (Bulté and Tezcan, 2019)
NLP	Natural language processing
NMT	Neural MT
PE	Post-editing (of MT)
QA	Quality assurance
RNN	Recurrent neural network
SGD	Stochastic gradient descent
TB	Terminology database (termbase)
TM	Translation memory
UI	User interface
WMT	Conference on Machine Translation (formerly Workshop on Statistical Machine Translation)
WYSIWYG	What you see is what you get

Chapter 1

Introduction

Computers have long become indispensable to professional translators. Since the introduction of word-processing software, the technologisation of translation has increased steadily with the emergence of translation memories (TMs) in the 1990s, the advent of online dictionaries and encyclopedias in the 2000s, and more recently with advances in the field of MT. Without a doubt, professional translation has become a form of human–computer interaction (HCI).

1.1 Key Problems

Nevertheless, professional translators uphold a sceptical attitude towards translation technology, as manifested in emotional discussions and slow adoption – especially with MT, the focus of this thesis. We attribute this to three key problems.

1.1.1 Quality

The quality of MT is underestimated by professional translators and overestimated by its developers. In a survey among the former, Cadwell et al. (2018) find that more than 80 % feel that 'MT can be ineffective for certain types of text' and 'produces output of poor quality for certain language pairs'. Poor quality is also a recurrent theme in discussions among professional translators on social media, where negative comments on MT outweigh positive comments by a 5:1 ratio (Läubli and Orrego-Carmona, 2017). While there can be no doubt that there are settings in which MT is not helpful, a growing body of studies on translator productivity – a proxy for MT quality – shows that in both lab and field experiments, the use of MT enables professionals to translate faster at no loss of quality. This conclusion has not only been drawn for many language pairs (e.g., Green et al., 2013) but also various domains and text types, from technical documentation (Plitt and Masselot, 2010) to marketing texts (Läubli et al., 2013) and even literature (Toral et al., 2018). Green et al. (2013) find that translators 'may have dated perceptions of MT quality that

do not account for the rapid progress in the field', and a study by Gaspari et al. (2014) shows that perceived effort in professionals who use MT can differ considerably from actual measurements.

MT researchers and developers, on the other hand, have downplayed the role of human translators for decades. Since the first MT experiments in the 1950s, it has been suggested that fully automatic high-quality MT (known as FAHQMT) was achievable within 'five, perhaps three years',[1] even if empirical assessments kept proving the opposite (e.g., Pierce et al., 1966; Krings, 1994). Exaggerated claims about MT quality – recently culminating in the assumption that MT had reached parity with professional human translation (Hassan et al., 2018) – have caused professionals to doubt technological progress in the field (Läubli and Orrego-Carmona, 2017), and may well be among the reasons for 'fear (e.g. of the unknown, of being replaced by a machine)' in more than 50 % of the translators surveyed by Cadwell et al. (2018).

Scientific evidence that MT quality is good enough to make professional translators more efficient, but not good enough to replace them (as we show in Chapter 3), calls for a focus on processes that combine the strengths of precision-oriented humans and recall-based machines. A prerequisite for convincing professional translators to participate in such processes is an adequate method to assess MT quality against the backdrop of human–machine parity claims.

1.1.2 Presentation

As professional translators revise MT output rather than create translations from scratch, their main activity shifts from writing to reading (do Carmo and Moorkens, 2020). In addition to the source text, translators need to parse and validate the machine-generated target text – the better the latter, the less time is devoted to writing in the overall translation process.

However, the computer-aided translation (CAT) tools available to translators are optimised for writing. CAT tools make two decades-old assumptions: they segment texts into sentences,[2] and arrange them in a spreadsheet-like view with source sentences on the left and target sentences on the right.

These assumptions seem to be influenced by the needs of translation memory (TM) and MT systems rather than human factors. According to the ISO standard on post-editing of MT output, texts are segmented to help computer applications, not translators: a segment is defined as a 'unit of *text* . . . produced for a computer application to facilitate translation' (ISO 18587:2017, 3.2.9). While sentence segmentation does facilitate the technical process of retrieving suggestions from TMs, it has been described as 'irritating' by translators

[1] See the IBM press release on the Georgetown-IBM experiment in 1954, available online at `https://www.ibm.com/ibm/history/exhibits/701/701_translator.html`

[2] Many CAT tools can be configured to operate at the level of paragraphs, but since retrieval rates from TMs are lower, sentence-level segmentation is more common.

since it results in an 'obstructed view of the text', making it difficult for them to focus on document-level cohesion (O'Brien et al., 2017).

The impact of text presentation on translator performance has, to the best of our knowledge, never been tested empirically. As for the placement of source and target text, it has been argued that a stacked (top-and-bottom) rather than left-and-right arrangement would reduce gaze shift, the time it takes translators to realign their line of sight to the relevant segment, and improve legibility (Green et al., 2014a). As translators write less and read more because they leverage MT suggestions, a user interface (UI) optimised for reading may lead to higher efficiency and satisfaction; and as MT systems are becoming able to consider document- rather than sentence-level context to produce translation suggestions (Junczys-Dowmunt, 2019; Popel et al., 2019), a UI that does not split texts into sentences may ease revision for document-level cohesion.

1.1.3 Adaptability

CAT tools and MT in particular are also characterised by a lack of means for translators to truly interact with machines (O'Brien, 2012). This is particularly evident in the revision of machine translated text, where systems do typically not respond to user feedback after an initial suggestion. Previous work has explored interactive MT (IMT), where humans and machines take turns in creating translations (Bar-Hillel, 1951; Langlais et al., 2000), and found that interactive forms of machine assistance can improve both translator efficiency and satisfaction (Green et al., 2014a).

Research on IMT has almost exclusively focussed on prefix decoding (Foster et al., 1997; Knowles and Koehn, 2016). Given the beginning of a translation produced by a user, an IMT system will provide suggestions for the next words or a completion of the current sentence in the target language. However, the system cannot suggest alternative translations for previous words or the beginning of a translation given its end (i.e., a suffix rather than a prefix) and thus enforces strictly unidirectional translation. As a consequence, translators cannot request MT for specific parts of a drafted sentence, such as a noun phrase they find difficult to translate.

An interaction paradigm that allows professional translators to use MT where they see fit may result in an increased sense of control (i.e., perceived agency; see Heer, 2019). Kay (1980) envisioned a form of human–machine collaboration where 'the translator has his say while the translation is under way', but forty years later, it would seem that technology dictates what translators can and cannot do, rather than vice versa. Translators have also voiced concerns about being primed by wrong or unnatural MT output (Cadwell et al., 2018), which an interaction paradigm where MT is only shown upon explicit request – in contrast to the always-on 'ghost text' in widely used predictive typing features – may alleviate. From a more technical perspective, moving away from prefix-decoding may also ease the inclusion of constraints that are known upfront, such as terminology from a repository that a translator (or their client) deems adequate for a given text.

1.2 Thesis Contributions and Outline

We first introduce the key methods, processes, and resources in professional human translation (Section 2.1), MT (Section 2.2), and mixed-initiative translation (Section 2.3).

Chapter 3 is focused on quality evaluation. Observing that MT researchers and users have different perceptions of MT quality, we reassess Hassan et al.'s (2018) finding of human–machine parity in Chinese to English news translation, and show that this finding is owed to weaknesses in best practices for MT evaluation: while professional translators find no significant difference in quality between Hassan et al.'s MT and human reference translations in a blind evaluation of isolated sentences, they show a significant preference for the latter in a blind evaluation of full documents. We synthesise our findings with concurrent work by Toral et al. (2018) to offer a set of recommendations for assessing strong MT systems in general and human–machine parity in particular: professional translators (rather than researchers or crowdworkers) should rate full documents (rather than isolated sentences), considering both fluency and adequacy. Reference source texts should be original (to avoid translationese), and human reference translations should not be heavily edited for fluency.

We then turn to presentation, the second key problem outlined above. We conduct semi-structured interviews on translation technology with 8 professional translators, and learn that visual context is one of three main areas where our interviewees see room for improvement in CAT tools (Chapter 4). To this end, we explore changes to fundamental design choices in the UIs of these tools in Chapter 5: in a controlled experiment with 20 professional translators, we test the impact of segmentation and orientation on text processing speed and accuracy. We find significant evidence that sentence-by-sentence presentation enables faster text reproduction and within-sentence error identification compared to unsegmented text, and that a top-and-bottom arrangement of source and target sentences enables faster text reproduction compared to a side-by-side arrangement. For revision, on the other hand, our results suggest that presenting unsegmented text results in the highest accuracy and time efficiency. Our findings have direct implications for best practices in designing CAT tools.

In Chapter 6, we focus on adaptability. We review several methods to incorporate prior knowledge – such as partial translations by a translator, terminology, or fuzzy matches – into MT output, and present Infix Generation, a multi-encoder alternative to constrained decoding with neural MT models. In contrast to prefix decoding (Knowles and Koehn, 2016), it allows the inclusion of a (possibly empty) suffix, and achieves a speedup of an order of magnitude over Grid Beam Search (Hokamp and Liu, 2017), a constrained decoding method with similar capacity.

We conclude by summarising the empirical findings, practical implications, and methodological suggestions of our work in Chapter 7, where we also discuss limitations and avenues for future research.

1.3 Publications

Significant portions of this thesis are based on work published during the author's time as a graduate student, as listed below.

Quality Assessment

- Läubli, Samuel, Rico Sennrich, and Martin Volk. 2018b. Has machine translation achieved human parity? A case for document-level evaluation. In *Proceedings of EMNLP*. Brussels, Belgium, pages 4791–4796.
- Läubli, Samuel, Sheila Casthilo, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020a. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research* 67:653–672.
- Fischer, Lukas and Samuel Läubli. 2020. What's the difference between professional human and machine translation? A blind multi-language study on domain-specific MT. In *Proceedings of EAMT*. Lisbon, Portugal, pages 215–224.

Translator Experience and Productivity

- Läubli, Samuel and David Orrego-Carmona. 2017. When Google Translate is better than some human colleagues, those people are no longer colleagues. In *Proceedings of Translating and the Computer*. London, UK, pages 59–69.
- Läubli, Samuel, Chantal Amrhein, Patrick Düggelin, Beatriz Gonzalez, Alena Zwahlen, and Martin Volk. 2019. Post-editing productivity with neural machine translation: An empirical assessment of speed and quality in the banking and finance domain. In *Proceedings of Machine Translation Summit XVII*. Dublin, Ireland, pages 267–272.
- Läubli, Samuel, Patrick Simianer, Joern Wuebker, Geza Kovacs, Rico Sennrich, and Spence Green. 2020b. The impact of text presentation on translator performance. Under review.

Reviews

- Läubli, Samuel and Spence Green. 2019. Translation technology research and human–computer interaction. In Minako O'Hagan, editor, *The Routledge Handbook of Translation and Technology*, Routledge, chapter 22, pages 370–383.

Software

- Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: A toolkit for neural machine translation. In *Proceedings of EACL*. Valencia, Spain, pages 65–68.
- Läubli, Samuel, Matthias Müller, Beat Horat, and Martin Volk. 2018a. mtrain: A convenience tool for machine translation. In *Proceedings of EAMT*. Alacant, Spain, page 357.

Chapter 2

Background

Translation is a complex task that involves natural language understanding and generation. This is challenging for both humans and machines, and while the latter are faster, only the former can provide a certificate of correctness, which is required in many applications. In this chapter, we summarise how language translation has been approached by specialised humans (Section 2.1) and by means of computer software (Section 2.2), and how the two have been supplementing each other in mixed-initiative settings (Section 2.3).

2.1 Professional Human Translation

Translation is one of the oldest professions in the world (Sager, 1994), and while its historical trajectory is beyond the scope of this thesis, it is vital to understand the tasks and processes involved in professional translation as practised today in order to identify meaningful applications of MT for professional translators. Sections 2.1.1 and 2.1.2 of our introduction are mostly based on international standards (ISO 17100:2015; ISO 18587:2017), which are primarily concerned with resources and processes. While they require that proper quality assessment processes be in place, they do not specify what these processes entail, and since translation quality assessment is one of the key problems addressed in this thesis (Section 1.1.1), Section 2.1.4 gives an overview of practical approaches to quality evaluation in professional translation workflows.

2.1.1 Resources

The two key resources in professional translation are skilled human workforce and technology. In terms of human resources, ISO 17100:2015 states that the people who perform translation tasks (Section 2.1.2) must have 'the required competences and qualifications'. Translation and revision require linguistic competences in the source and target

language, and – perhaps less obvious to readers with a background in MT – cultural, technical, and domain competences, according to the ISO standard. Reviewing and proofreading, on the other hand, are monolingual tasks that do not require competences in the source language.

In order to qualify as a professional translator or reviewer according to ISO 17100:2015, individuals need either a university degree in translation, a university degree in any other field plus two years of full-time professional experience in translation, or five years of full-time professional experience in translation. However, this requirement is subject to discussion. Research has shown that people lacking these qualifications can, in some contexts, produce translations that are 'as good as their professional counterparts' (Orrego-Carmona, 2016). Many translation agencies subcontract translation tasks to workers who do not meet the ISO qualifications for reasons of cost efficiency (Olohan, 2007). Individuals who do meet these qualifications, on the other hand, often organise in professional associations to distinguish themselves from lay people, especially in countries where 'translator' is not a protected professional title.[1]

ISO 17100:2015 also requires translation service providers to have a technical infrastructure in place. The infrastructure must comprise hardware and software for communication, project management, and knowledge acquisition – essentially computers with internet access, long indispensable to professional translators (O'Brien, 2012). Pertaining to the possibly sensitive nature of content to be translated, the infrastructure must ensure 'safe and confidential handling … of all relevant data and documents' (ISO 17100:2015, 3.2.a). It must also provide specialised software for leveraging and curating translation-related language resources (ISO 17100:2015, 3.2.d). Typically, this entails translation workbenches – so-called CAT tools, described further in Section 2.3.2 – as well as translation memories (TMs) and terminology databases (termbases, TBs).

TMs store translated segments, either sentences or paragraphs. If a document to be translated contains a source segment for which a translation is found in the TM, this translation can be inserted into the target document and adapted by the translator as needed. A translation found in the TM is referred to as exact or fuzzy match if its source text is exactly the same or differs to some degree from the source text in the current document, respectively. The degree to which the current source segment and the source segment in the TM overlap is termed fuzzy match score, and typically calculated as

$$\left(1 - \frac{\text{token-based minimum edit distance}}{\text{\# tokens in longer segment}}\right) \times 100, \tag{2.1}$$

with weighted costs for token classes (regular word, punctuation, number, tag, etc.). These costs are configurable in most CAT tools.

TBs, on the other hand, store translations of terminology. Terms are typically stored in their base form, and may be annotated with metadata such as descriptions, usage examples, or linguistic features (such as part-of-speech). CAT tools will look up the words

[1] Such as Switzerland, Germany, or the United States.

in a segment being translated in the TB that is associated with a translation job, if any, and display any matches in a dedicated window (see the Term Recognition window in Figure 2.12). The motivation is to ensure that specific words are translated in a specific way, e.g., when different translators work on a domain-specific translation concurrently or over time.

2.1.2 Translation Tasks and Processes

The professional translation workflow according to ISO 17100:2015 is organised in three processes: pre-production, production, and post-production. The goal is to translate source language content[2] provided by a client into target language content. To give just two examples, the source language content of a translation job could be a written contract or a set of TV subtitles.

Pre-production

The pre-production process involves both business-related and technical tasks. The former include handling the client's enquiry, assessing feasibility, producing a quote, and negotiating an agreement (contract).

Once an agreement is reached, the source language document[3] is prepared for translation. In particular, the document is typically normalised into an electronic format that strips formatting from text (using software described in Section 2.3.2). The text is then enriched with suitable translations from previous jobs. Suitable translations are retrieved by comparing the text to translations in TMs and possibly TBs (Section 2.1.1), where normalisation ensures that formatting does not bias the comparison. If the source document, for example, contains the sentence *The dog she loves* in green font, the font colour should not prevent the retrieval of a translation for *The dog she loves* in blue font from a previous job. The aim of packaging suitable translations from previous jobs with the current job is to reduce research and translation effort on the translator's part in the production process (see below).

All pre-production tasks can be assumed by a project manager. Large service providers will typically delegate some business-related tasks to sales or administrative personnel. In smaller settings, some or all of the technical tasks may be delegated to a translator who is also involved in the production process.

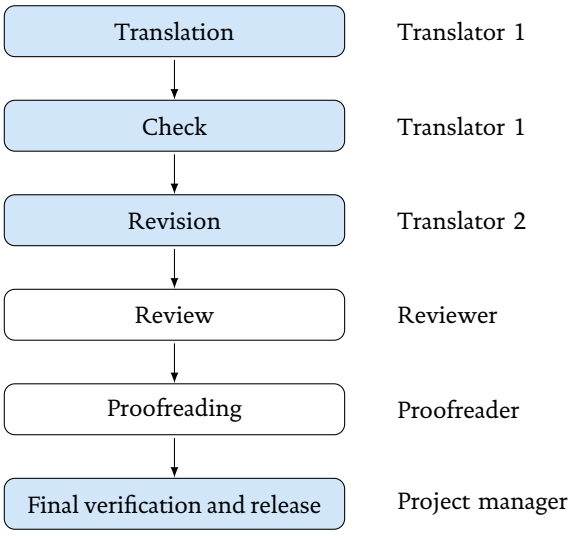


Figure 2.1: Translation workflow: production process according to ISO 17100:2015. Stages in white boxes are optional.

Production

After pre-production, the project manager initiates and orchestrates the production process. This process, depicted in Figure 2.1, involves four required and two optional tasks.

Production starts with translating the source document into the target language. The translator who performs this task is also required to perform a check of the produced translation, i.e., self-revision for errors and compliance with any relevant client specifications. The translator contacts the project manager in case of uncertainties, who in turn checks back with the client as needed.

The next task in the production process is revision. The reviser must have bilingual competences (Section 2.1.1) as their task is to examine the source language content against the translation produced in the previous step, and either correct errors or suggest corrections to the original translator. The reviser must be a person other than the translator (ISO 17100:2015, 5.3.3).

Revision is followed by two optional tasks exclusively focussed on the target language content: review and proofreading. The goal of a review is to assess the translation's domain and text type accuracy. The goal of proofreading is to reveal linguistic defects. Both reviewers and proofreaders are asked to either suggest or directly implement corrections.

A final verification by the project manager concludes the production process. If the veri-

[2] Content is 'anything representing meaningful information or knowledge' (ISO 17100:2015, 2.3.1).

[3] A document is defined as 'information and its supporting medium' (ISO 17100:2015, 2.5.2). A translation job can contain multiple documents.

fication does not reveal any problems, the translation is delivered to the client.

Post-production

ISO 17100:2015 requires translation service providers to have a process in place to handle client feedback, assess client satisfaction, and redeliver the adjusted translation if corrections are needed. The ISO standard encourages that all feedback be shared with everyone involved in the translation workflow.

Lastly, the translation project is archived in compliance with legal and contractual requirements. These requirements may, for example, include taking appropriate measures for data protection (ISO 17100:2015, 6.2).

2.1.3 Post-editing Tasks and Processes

The main difference between post-editing and the 'traditional' translation workflow, as described in the previous section, is the use of MT. While only some segments of the source text will be paired with translations from TMs (exact or fuzzy matches) in the traditional workflow, MT will be used to pre-translate the entire source text in the pre-production process.

Whereas the production process involves at least two bilingual individuals in the traditional workflow, the ISO standard on post-editing (ISO 18587:2017) mentions no such requirement. The translation task in Figure 2.1 is essentially replaced by MT, and a professional human translator then checks and revises the MT output as much as needed for information gisting (light post-editing) or publication (full post-editing).

Post-editing enables faster translation at the same or slightly better quality in many languages and domains (Section 2.3), but requires more competences on part of the human translator: in addition to a translation degree or equivalent experience in traditional translation (Section 2.1.1), post-editors require advanced technical competences – such as 'a general knowledge of MT technology and a basic understanding of common errors that an MT system makes' – and 'the ability to provide structured feedback on frequently recurring errors in the MT output' (ISO 18587:2017, 5.3). Despite the higher requirements, post-editing currently has a lower reputation than traditional translation among human professionals (Daems, 2016, p. 159).

2.1.4 Evaluation

How to tell whether a translation is good or bad is one of the most important and one of the most difficult questions asked in connection with translation. From a user's or client's perspective, quality is important because it determines whether a translation is fit for purpose; from a professional translator's perspective, quality is important because it justifies and determines the price a client is willing to pay for their service. The inherent difficulty

in assessing translation quality is rooted in the fact that there is no single correct translation for any source text because a target language offers several valid ways of expressing its meaning.[4]

The definition and methods used to assess translation quality differ considerably in theory and practice. In academic contexts (i.e., the field of translation studies), there is consensus that any sort of quality assessment presupposes a theory of translation, and the focus is on constructing and debating this theory rather than offering practical methods for evaluation. Translation quality is studied (described) rather than measured (put to numbers), and an in-depth discussion is beyond the scope of this thesis. For an overview, see House (2013).

In practice (i.e., the translation industry), buyers and providers of translation services are not only concerned with the quality of the translated material, but also the quality of the transaction (Gouadec, 2010). The inherent assumption is that if professional translators strictly follow a set of relevant procedures, such as those defined in the ISO standards on translation (Section 2.1.2) and post-editing (Section 2.1.3), the risk of insufficient quality is minimised. However, the ISO standards do not define any metrics for quality assessment per se. If a reviser finds an error in a translation, ISO 17100:2015 stipulates a process of correcting or reporting it back to the translator; if a client finds an error in a translation, ISO 17100:2015 stipulates a process for reporting it back to the project manager, who in turn coordinates with the translator and/or reviser and delivers the adjusted translation back to the client.

In some use cases, however, it is desirable to explicitly measure how good a translation is. Consider the benchmarking of individual translators, e.g., to assess if a translator produces translations of sufficient quality in a specialised domain, or the benchmarking of translation agencies, e.g., to choose a new provider or monitor translation quality over time with existing providers at large institutions.

To that end, human raters – typically revisers with a background in professional translation – evaluate a translation (as a whole, or a random sample thereof) by identifying errors according to a quality standard. LISA QA, the first quality standard to gain widespread adoption in the translation industry, defines 20–123 error types and three severity levels: minor, major, and critical. SAE J2450, originating from the automotive industry, uses fewer error types and only two severity levels: minor and major. In contrast to LISA QA, SAE J2450 focusses exclusively on linguistic quality (i.e., no style and formatting, etc.). More recently, a joint academia-industry initiative has proposed the Multidimensional Quality Metrics (MQM) framework. This framework allows the definition of custom quality metrics by choosing a subset of (weighted) error categories, and provides mappings to SAE J2450. Evaluations are typically conducted with standalone software (Figure 2.2) or within translation workbenches (Section 2.3.2), which record the number of errors flagged by a rater and, depending on the metric, compute a final score by combin-

[4] Variation can arise from linguistic phenomena like synonymy and extra-linguistic phenomena such as gender associations, to name just a few. See Assis Rosa (2012) for further examples and discussion.

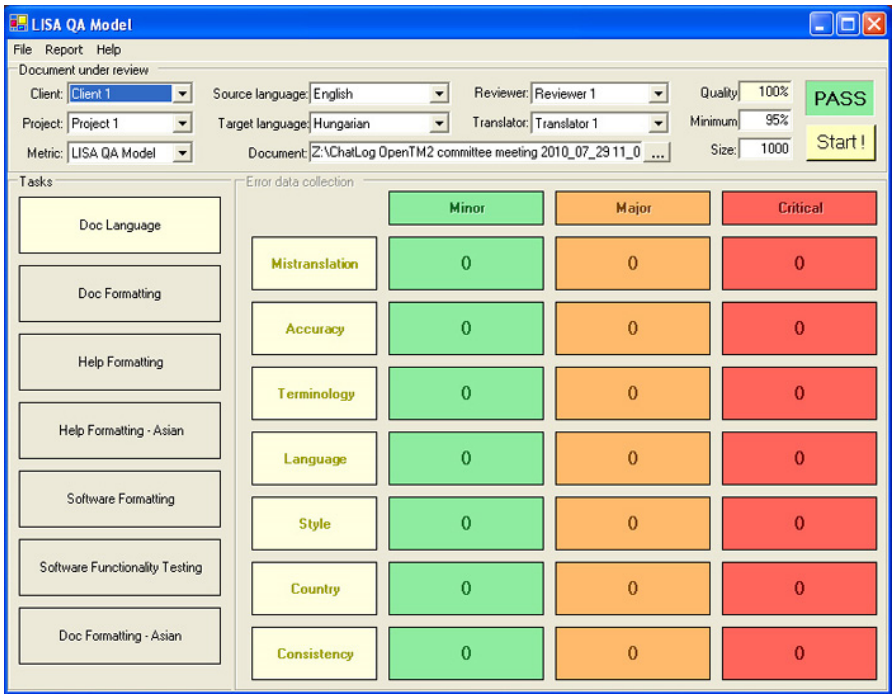


Figure 2.2: The LISA QA 3.1 metric system (original implementation). Screenshot retrieved from `https://sites.miis.edu/sscottodantuono/files/2019/05/xscfsqty.jpg`.

ing error counts and weights. While this methodology is similar to human error categorisation in MT (Section 2.2.3), we note that raters evaluate translated texts as a whole or coherent samples thereof (e.g., the first three paragraphs).

2.2 Machine Translation

In this section, we introduce the way in which text can be translated from one language into another by means of machines rather than (professional) human translators. We start with a brief overview of the historical context, and then focus on data-driven approaches, i.e., machine learning methods to learn translation models from large quantities of translated text (the so-called training data) and then produce new translations by applying these models to monolingual text in the source language (referred to as test data in experimental settings). The focus is on neural methods (Section 2.2.2), which are the basis for the Chinese to English system we evaluate in Chapter 3, and the infix generation model we introduce and evaluate in Chapter 6. The latter is intended for use in mixed-initiative settings where humans and machines take turns in translating text, a paradigm we describe further in Section 2.3.

The mathematical details in this section will be relevant for readers who wish to study Chapter 6 of this thesis. Readers interested in Chapters 3–5 may want to skip them.

2.2.1 Historical Context

Efforts to translate natural language with machines date back to the Cold War, when the interest in translating Russian texts into English as quickly as possible led to major research investments in the United States from the late 1940s to the early 1960s. At the time, research centred around word replacement methods based on bilingual dictionaries and syntactic rules. These resources were handcrafted and only covered small fractions of language pairs, but it was expected that the approach would soon allow for broader coverage (e.g., Hutchins, 1997). A system resulting from cooperative research by IBM and the Georgetown Univeristy in 1954, for example, used 250 lexical items and 6 syntactic rules for English to Russian translation. Nevertheless, it was described as 'the culmination of centuries of search by scholars for a "mechanical translator"' in a front-page article in the New York Times, which also stated that 'there are no foreseeable limits to the number of words that the device can store or the number of languages it can be directed to translate' (Plumb, 1954).

While rule-based methods did not prove successful (Section 2.3.1), storage and computing capacity were indeed decisive factors for subsequent approaches. Seminal work by Brown et al. (1988, 1993) at IBM suggested to cast language translation as a mathematical problem: the probability of translating a source sentence x to a target sentence y is defined as $P(y|x)$, and decomposed into two statistical models – a translation model $P(x|y)$ and a language model $P(y)$ – by applying Bayes' rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{2.2}$$

$$\propto P(x|y)P(y). \tag{2.3}$$

Brown et al. (1993) suggested five algorithms with increasing complexity to learn word-level translation models from a set of translated sentences, a parallel corpus referred to as training data, by means of expectation–maximisation training (Dempster et al., 1977). The basic idea is to start off with the assumption that any source word can be translated into any target word with equal probability, and then refine the translation probability of every source–target word pair by exploiting the observation that words which are translations of each another (such as *dog* and *Hund*) tend to co-occur more frequently in translated sentences than word pairs which are not (such as *dog* and *Gitarre*).

$P(y)$ is a language model, capturing the likelihood of word sequences in a given language. A sound language model for English, for example, would assign a higher probability to *she loves her dog* than *she love her dog*. By relying on the chain rule, the probability of a word sequence can be defined as

$$P(y_1, \ldots, y_T) = \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}), \tag{2.4}$$

where y_t is the t-th word in the sequence of length T.[5] Vocabulary V defines the set of valid types. All types can occur at all positions in the sequence, but some combinations will be more likely than others (see above). The likelihood of such combinations can be derived from monolingual text. However, even very large text collections will not contain all possible (or even all valid) combinations, so some form of generalisation is needed.

Count-based methods rely on the Markov assumption. Rather than conditioning the probability of y_t on all previous words $y_1, \ldots, y_{t-1}$, the context is limited to N–1 previous words in a model of order N,

$$P(y_1, \ldots, y_T) = \prod_{t=1}^{T} P(y_t | y_{t-N+1}, \ldots, y_{t-1}), \tag{2.5}$$

and the likelihood of each N-gram $P(y_t | y_{t-N+1}, \ldots, y_{t-1})$ is derived by counting how often the word y_t follows the context (or history) $y_{t-N+1}, \ldots, y_{t-1}$ in the training data:

$$P(y_t | y_{t-N+1}, \ldots, y_{t-1}) = \frac{\text{count}(y_{t-N+1}, \ldots, y_{t-1}, y_t)}{\text{count}(y_{t-N+1}, \ldots, y_{t-1})}. \tag{2.6}$$

Methods such as interpolated Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998) are used for more robust parameter estimation, and to avoid zero-probabilities with N-grams not contained in the training data.

Given a trained translation and language model, the translation of an unseen sentence x becomes a search problem: out of all possible sentences in the target language, we are looking for the target sentence $\hat{y}$ which maximises the joint model probabilities, i.e.,

$$\hat{y} = \arg\max_y P(x|y)P(y). \tag{2.7}$$

Similar methods had been theorised decades earlier (Weaver, 1947), but were abandoned due to the need for large volumes of translated text, and the computational means to process them, for reliable parameter estimation.

Subsequent approaches were based on log-linear models, which allowed the weighting and incorporation of additional model components. In this framework, $\hat{y}$ is the target sentence that maximises the log-linear combination of F arbitrary model component scores ϕ – referred to as features – with a weight vector $\mathbf{w}$:

$$\hat{y} = \arg\max_y \sum_{i=1}^{F} w_i \phi_i. \tag{2.8}$$

[5] We use log probabilities in practice for numerical stability and efficient computation. Regular and log probabilities are used synonymously throughout this thesis.

The weights $\mathbf{w}$ are typically optimised on a held-out development set, i.e., data that was not used to train the features ϕ. Equation 2.7 can be reformulated as a log-linear model with two equally weighted features, i.e.,

$$\hat{y} = \arg\max_{y} \left(\tfrac{1}{2} \log P(x|y) + \tfrac{1}{2} \log P(y) \right). \tag{2.9}$$

The inclusion of additional features led to substantial improvements in translation quality. In particular, phrase-based statistical MT (Koehn et al., 2003) emerged as the dominant paradigm in the middle of the first decade of the 2000s, and enabled successful applications in commercial settings (e.g., Flournoy and Duran, 2009; Plitt and Masselot, 2010). To counteract the strong independence assumption between features that word-based models make, Koehn et al. (2003) use multi-word units – so-called phrases – in addition to single words in the translation model,[6] and add an additional model to account for differences in word order between the source and target language (Koehn et al., 2005). All models are trained independently of each other using bilingual (or, in the case of the language model, monolingual) training data, and their weights are typically optimised using minimum error rate training (Och, 2003).

2.2.2 Neural Machine Translation

The log-linear model suffers from two major weaknesses. First, dependencies between features cannot be modelled. For example, it is not possible to give more weight to the language model if the phrase translation score is low. Second, the relationship between a feature score and its impact on the overall score cannot be non-linear. Considering Equation 2.9, we could give more weight to the language model score overall – but not disproportionately more if it exceeds or disproportionately less if it falls below a certain threshold, since the weight of a feature w_i cannot depend on its score ϕ_i (see Equation 2.8).

The strong independence assumption that the log-linear model makes is the primary motivation for end-to-end learning. Rather than combining models whose parameters are estimated independently of each other, end-to-end approaches use a single model in which all parameters are jointly optimised. In this section, we introduce a number of these models and describe their application to language modelling and translation.

Neural Feedforward Networks

Neural feedforward networks differ from linear models in two ways: the output is derived from one or more vectors of intermediate results rather than directly from the input

[6] The terms 'model' and 'feature' are used somewhat interchangeably in the literature. Strictly speaking, a model (such as a translation model) can consist of multiple features (such as direct and inverse translation probability).

values, and a non-linear transformation is applied to each intermediate result. The vectors of intermediate results and the non-linear transformation are referred to as hidden layers and non-linear activation function, respectively.

Figure 2.3 contrasts a linear model with a feedforward network. Both transform an input vector $\mathbf{x}$ of size $S = 2, \mathbf{x} = (x_1, x_2)$, to a single-valued output y. The linear model does so by multiplying $\mathbf{x}$ with the weight vector $\mathbf{w}$, i.e.,

$$y = \mathbf{w}\mathbf{x} + b = \sum_{i=1}^{S} w_i x_i + b, \tag{2.10}$$

where b is a bias term. In the feedforward network, the input $\mathbf{x}$ is first transformed into an intermediate result $\mathbf{h}$. Graphically speaking, each neuron of the hidden layer $\mathbf{h}$ is connected to a bias node and each neuron of the previous layer, the input $\mathbf{x}$ in this case. The connection between any two neurons x_i and h_j is associated with a weight w_{ji} in the weight matrix $\mathbf{W}$, and $\mathbf{h}$ is computed as

$$\mathbf{h} = g(\mathbf{W}\mathbf{x} + \mathbf{b}), \text{ or } h_j = g\left(\sum_{i=1}^{S} w_{ji} x_i + b_i\right) \text{ for every } h_j \in \mathbf{h}, \tag{2.11}$$

where $\mathbf{b}$ is a vector of bias weights and g is a non-linear activation function. Typical choices for g are

- the logistic function $\sigma(x) = \frac{1}{e^{-x}}$,
- the hyperbolic tangent $\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, and
- the rectified linear unit $\text{relu}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$.

Neural feedforward networks can approximate arbitrary functions, and a higher number of parameters (through more hidden layers and/or neurons per hidden layer) can lead to better approximation. The parameters are optimised via gradient-based learning on a set of training examples (the training data), as described in (Goodfellow et al., 2016, inter alia). In the case of language modelling and translation, the training data consists of monolingual or translated text, respectively, and the goal is to optimise the model parameters so as to minimise perplexity or maximise a metric like BLEU (Section 2.2.3) on the training data. All of the architectures described in the following sections are sensitive to weight initialisation and the choice of hyperparameters, and we refer the reader to the respective publications for details.

Neural Language Models

Non-linear models such as neural feedforward networks can be used in language modelling to overcome a fundamental weakness of count-based language models, which treat

(a) Linear model.

(b) Neural feedforward network.

Figure 2.3: A linear model (a) vs. a neural feedforward network (b).

words as discrete units with no inherent relationship to one another (Section 2.2.1, Equation 2.6). The basic requirement here is that a language model should be able to infer that the word *dog* is similarly likely as a continuation of *she loves her* and *he adores his* due to the similar semantic and grammatical roles of the words *she* and *he*, *likes* and *adores*, etc.

To this end, Bengio et al. (2000, 2003, 2006) couple the learning of an N-gram language model (Equation 2.5) with the learning of numerical word representations. Each word in the vocabulary V is associated with a real-valued vector in an m-dimensonal space $\mathbb{R}^m$, where m is much smaller than the vocabulary size $|V|$. Similar words are expected to have similar representations (i.e., they are 'close to each other' in $\mathbb{R}^m$), and thus small changes in word sequences – such as replacing *loves* with *adores* – are expected to induce small changes in probability.

The architecture consists of two parts: a mapping $\mathbf{C}$ from each word $w \in V$ to a real-valued vector $\mathbf{c}_w$, implemented as a $|V| \times m$ matrix of free parameters;[7] and a probability function, implemented as a feedforward network, which maps an input sequence x for a position t in a text to a conditional probability distribution over all words $w \in V$ for w_t. The input sequence is the concatenation of N–1 word vectors,

$$\mathbf{x}_t = (\mathbf{c}_{w_{t-N+1}}, \ldots, \mathbf{c}_{w_{t-1}}), \tag{2.12}$$

and the vector $\mathbf{y}_t$ of unnormalised log-probabilities for each word at position t is computed as

$$\mathbf{y}_t = \mathbf{b} + \mathbf{W}\mathbf{x}_t + \mathbf{U}\tanh(\mathbf{d} + \mathbf{H}\mathbf{x}), \tag{2.13}$$

where $\mathbf{b}$ and $\mathbf{d}$ are biases for the output and hidden layer; $\mathbf{U}$, $\mathbf{W}$, and $\mathbf{H}$ are the hidden-to-output, input-to-output,[8] and input-to-hidden layer weights, respectively. Normalised log-probabilities are obtained by applying the softmax function to $\mathbf{y}$, such that

[7] now referred to as word embedding matrix

[8] The residual input-to-output connections are optional. Bengio et al. (2003) obtain slightly better results without them, but note that the training takes twice as long to converge.

$$P(w_t|w_{t-N+1},\ldots,w_{t-1}) = \frac{\exp(\mathbf{y}_t^{w_t})}{\sum_{w\in V}\exp(\mathbf{y}_t^{w})}. \tag{2.14}$$

Bengio et al.'s (2003) method outperformed two count-based baselines on two standard English corpora well over a decade ago, but count-based methods remained the de-facto standard in many applications, including MT (e.g., Durrani et al., 2014). One reason was that training feedforward networks of reasonable size was 'a significant challenge' with the computational resources available at the time (Bengio et al., 2003).

Recurrent Neural Language Models

While improving generalisation to unseen word sequences, feedforward neural language models still operate on fixed-length inputs (Equation 2.12), i.e., the probability of a word depends on exactly N–1 previous words. This assumption does not hold in natural languages. Consider the following example:

(i) *the dog she loves most is called Fido*

(ii) *the dog she loves most are called Fido*

A model of order N=3 would fail to punish the wrong subject-verb agreement in the main clause of (ii), simply because it would decompose the sentence into isolated three-word sequences (trigrams) to compute its overall probability. Ideally, the probability of any word in the sequence would depend on all previous words (Equation 2.4).

Predicting the next word from any number of previous words requires an architecture that can handle variable-length input, and the first successful approaches to large-scale neural language modelling (Mikolov et al., 2010, 2011) relied on recurrent neural networks (Elman, 1990).

Recurrent neural networks read sequences of input vectors – such as numerical word representations in the case of language modelling – one by one, i.e., in timesteps. The hidden layer $\mathbf{h}$ acts as a memory cell. It encodes the current input $\mathbf{x}_t$ with the network's hidden layer state of the previous timestep $\mathbf{h}_{t-1}$. Mikolov et al. (2010) use two sets of weights $\mathbf{U}$ and $\mathbf{W}$ to transform these inputs, and compute the hidden layer state at timestep t as

$$\mathbf{h}_t = g(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}), \tag{2.15}$$

where g is a non-linear activation function. A softmax output layer (Equation 2.14) returns probabilities for all words in the vocabulary, using an additional weight matrix $\mathbf{V}$:

$$\mathbf{y}_t = \text{softmax}(\mathbf{V}\mathbf{h}_t). \tag{2.16}$$

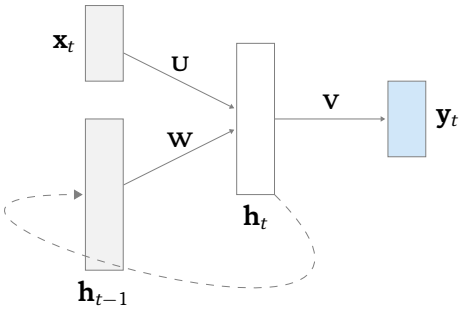


Figure 2.4: Recurrent neural network.

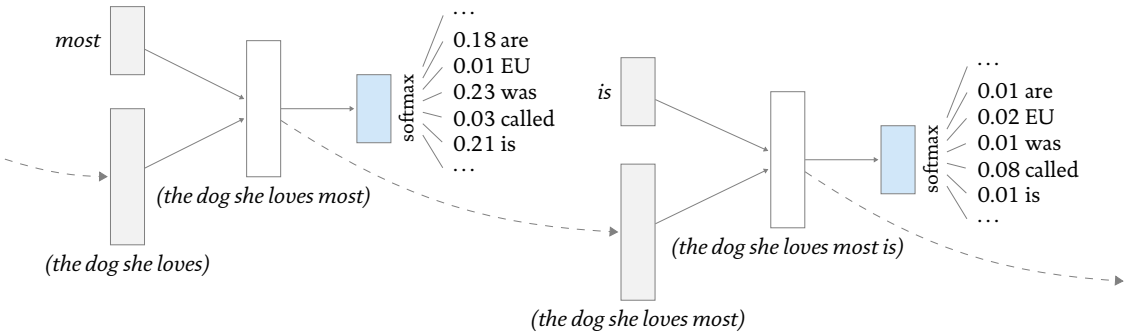


Figure 2.5: Scoring with a recurrent neural language model (example).

The architecture is depicted in Figure 2.4.

Figure 2.5 illustrates the scoring of example (i) above with a recurrent neural language model. At timestep t_5, the model has already processed the inputs *the* (at t_1), *dog* (t_2), *she* (t_3), and *loves* (t_4). The network's state at t_4, $\mathbf{h}_4$, is a numerical representation of these inputs, depicted as *(the dog she loves)* in the figure. We now input the next word, *most*, and combine it with this representation to form the network's new state at t_5, $\mathbf{h}_5$, as defined in Equation 2.15. After the full forward pass through the softmax layer, we obtain a probability distribution over all words in the vocabulary. At timestep t_5, we are interested in the probability of *is* given *the dog she loves the most*, and according to the network's output, $P(is|the\ dog\ she\ loves) = 0.21$. We repeat this process at t_6 by inputting the next word *is* and the current context vector $\mathbf{h}_5$, and obtain $P(called|the\ dog\ she\ loves\ is) = 0.08$. Note that we use the same network for all forward passes until we reach the end of the sequence to be scored; the input, hidden, and output layers in Figure 2.5 are just duplicated, or 'unrolled', for the sake of illustration.

Mikolov et al.'s (2010) recurrent neural language model can be trained using backpropagation with stochastic gradient descent (SGD). However, training with long input sequences suffers from the vanishing gradient problem. The gradient of the network's error function gets scaled when propagated back through one of its cells, and since the scaling factor is typically greater or smaller than one for relevant cases, the gradients

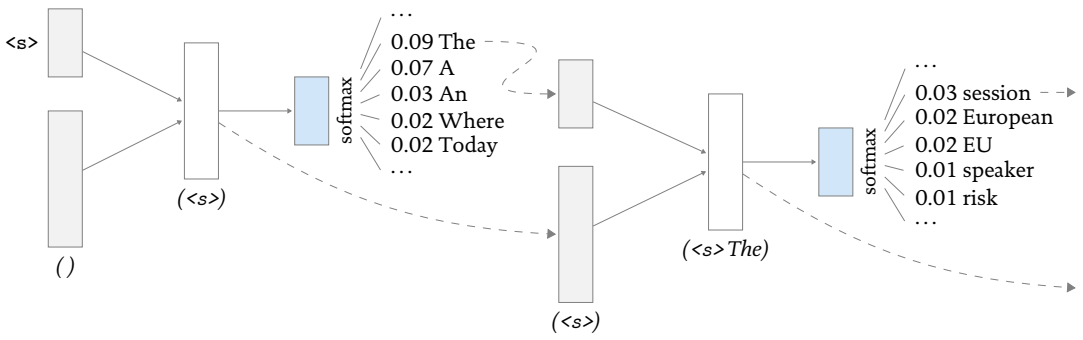


Figure 2.6: Sampling with a recurrent neural language model (example).

approach positive or negative infinity over time. The use of long short-term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) stabilises the gradients and leads to better performance (Sundermeyer et al., 2012).

In addition to scoring, recurrent neural language models can be used to generate word sequences that resemble their training data. This process is referred to as sampling. Recall from Equation 2.14 that given the vector representation of an input word and a context of previous input words, a recurrent neural language model returns a probability distribution over all words in the vocabulary. In the first timestep t_1, the input word is a special symbol to mark the beginning of the sequence, often represented as <s>, and the context of previous input words is typically set to zero. The first forward pass then generates the probability distribution for the first word in the sequence. In scoring, we would look up the probability of the first word in the sentence to be scored, and make this word the input in the next timestep (Figure 2.5). In sampling, we take the word with the highest probability[9] $\hat{y}_t$ according to the distribution generated in the current timestep t instead:

$$\hat{y}_t = \arg\max_{y \in V} p(y|y_1, \ldots, y_{t-1}). \tag{2.17}$$

$\hat{y}_t$ is added to the output sequence, and used as input in the next timestep. This procedure, illustrated in Figure 2.6, is repeated until the most probable next word is </s>, the special symbol to mark the end of a sequence.

Recurrent Sequence to Sequence Models

The first successful neural MT systems took advantage of the ability of recurrent neural language models to generate word sequences. The architecture proposed by Sutskever et al. (2014) is, in essence, a language model of the target language with access to a numerical representation of the source sentence to be translated. Let $x = x_1, \ldots, x_S$ be the source sentence and $y = y_1, \ldots, y_T$ its translation, noting that their lengths S and T may differ.

[9] We may use ancestral sampling or a similar strategy to allow for some form of variation, i.e., different outputs when sampling from the same trained language model.

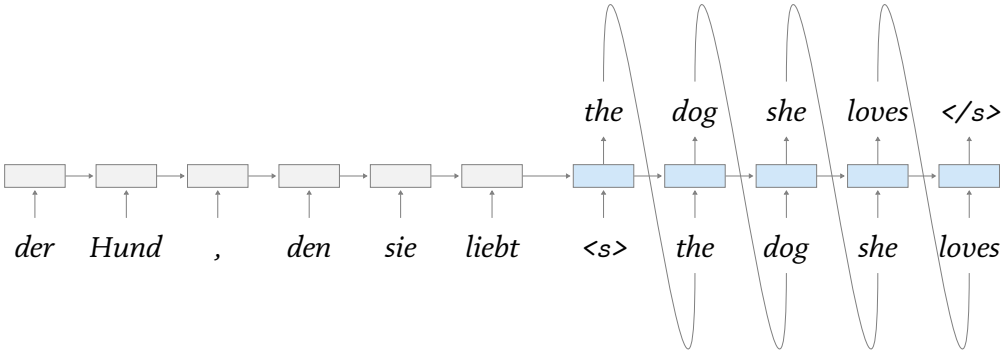


Figure 2.7: Neural sequence to sequence model (example).

By applying the chain rule (Equation 2.4), the probability of the translation is defined as

$$P(y) = \prod_{t=1}^{T} P(y_t | y_1, \ldots, y_{t-1}, \mathbf{c}). \tag{2.18}$$

The representation of the source sentence $\mathbf{c}$ is the last hidden state $\mathbf{h}_S$ of an LSTM f, the encoder network, which has sequentially consumed all words of the source sentence:

$$\begin{aligned} \mathbf{h}_s &= f(\mathbf{x}_s, \mathbf{h}_{s-1}); \\ \mathbf{c} &= \mathbf{h}_S. \end{aligned} \tag{2.19}$$

The words of the target sentence are then generated using another LSTM g, the decoder network. The procedure is the same as in language model sampling (Equation 2.18, Figure 2.6), except that $\mathbf{c}$ is used to initialise the decoder's context vector. The first word in the target sentence is thus not only determined by what typically occurs at the beginning of any sentence in the target language, but also by all words of the source sentence. Since the word generated at each timestep is reflected in the hidden state at the next timestep, each following target word is conditioned on all source words as well as well as all previously generated target words (Figure 2.7).

Sutskever et al. (2014) show that reversing the input order of the source words, as well as using multiple hidden layers in the encoder and decoder, beam search, and model ensembling, leads to better performance. Their approach was the first to outperform a phrase-based MT baseline (described in Cho et al., 2014b) on a large-scale task with a fully neural translation system, and came close to state-of-the-art performance on the 2014 WMT English to French news translation task through rescoring the output of the aforementioned baseline system (Sutskever et al., 2014).

Nevertheless, the sequence-to-sequence model introduced by Sutskever et al. (2014) suffers from two major limitations. First, the vocabulary size – the number of word forms that the encoder can consume and the decoder can generate – is limited. Due to sparse training data and the computational expense of the softmax operation (Equation 2.14), Sutskever et al. (2014, inter alia) replace rare words with a special symbol for unknown words (`<unk>`). This limitation can be circumvented by splitting words into subword units (Sennrich et al., 2016b). Second, the model compresses source sentences of any length into a fixed-length vector ($\mathbf{c}$), such that performance deteriorates with long input sentences (see also Cho et al., 2014a).

Attention

Bahdanau et al. (2015) extend the encoder introduced in the previous section such that the input sentence is not compressed into a single fixed-length vector (Equation 2.19),

but a fixed-length vector $\mathbf{h}_s$ for every input word $\mathbf{x}_s$. The encoder consists of two RNNs:[10] $\overrightarrow{f}$ reads all input words from left to right, and $\overleftarrow{f}$ reads all input words from right to left, such that

$$\begin{aligned}\overrightarrow{\mathbf{h}}_s &= \overrightarrow{f}(\mathbf{x}_s, \overrightarrow{\mathbf{h}}_{s-1}), \text{ and} \\ \overleftarrow{\mathbf{h}}_s &= \overleftarrow{f}(\mathbf{x}_s, \overleftarrow{\mathbf{h}}_{s+1}),\end{aligned} \tag{2.20}$$

and $\mathbf{h}_s$ is the concatenation of the forward and backward hidden states at timestep s:

$$\mathbf{h}_s = \left[\overrightarrow{\mathbf{h}}_s; \overleftarrow{\mathbf{h}}_s\right]. \tag{2.21}$$

As a result, each context vector $\mathbf{h}_s$ encodes the current word ($\mathbf{x}_s$) in the context of all preceding ($\mathbf{x}_1, \ldots, \mathbf{x}_{s-1}$) and all following words ($\mathbf{x}_{s+1}, \ldots, \mathbf{x}_S$) in the input sentence.

The probability for each output word, conditioned on all input words and all previously generated output words, is computed using a decoder RNN g,

$$p(y_t|y_1, \ldots, y_{t-1}, x_1, \ldots, x_S) = g(\mathbf{y}_{t-1}, \mathbf{s}_t, \mathbf{c}_t), \text{where} \tag{2.22}$$

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{y}_{t-1}, \mathbf{c}_t). \tag{2.23}$$

In contrast to Sutskever et al.'s (2014) model, the context vector $\mathbf{c}$ is computed anew at every timestep in the decoder, reflecting the relative importance of each input word to produce the current output word. An example is shown in Figure 2.8, where the singular determiner *der*, the singular noun *Hund*, and the third-person present verb *heisst* are more important for producing the English output word *is* (rather than *are* or *was*) than the other words in the German input.

The relative weighting of encoded inputs, referred to as attention mechanism, is realised by computing a weighted sum over the sequence of encoder outputs $[\mathbf{h}_1, \ldots, \mathbf{h}_S]$:

$$\mathbf{c}_t = \sum_{s=1}^{S} \alpha_{ts}\mathbf{h}_s, \text{ where } \alpha_{ts} = \frac{\exp(\mathrm{e}_{ts})}{\sum_{k=1}^{S}\exp(\mathrm{e}_{tk})}, \text{ and } \mathrm{e}_{ts} = a(\mathbf{s}_{t-1}, \mathbf{h}_s). \tag{2.24}$$

The alignment model a scores how well the input words around position s match the output word at position t, and is implemented as a feedforward network (Page 16). It is trained jointly with the encoder and decoder.

Bahdanau et al. (2015) show that this sequence to sequence model with attention achieves better performance than the model described in the previous section (Sutskever

[10] Bahdanau et al. (2015) used GRU (Cho et al., 2014b) rather than LSTM cells to avoid vanishing gradients.

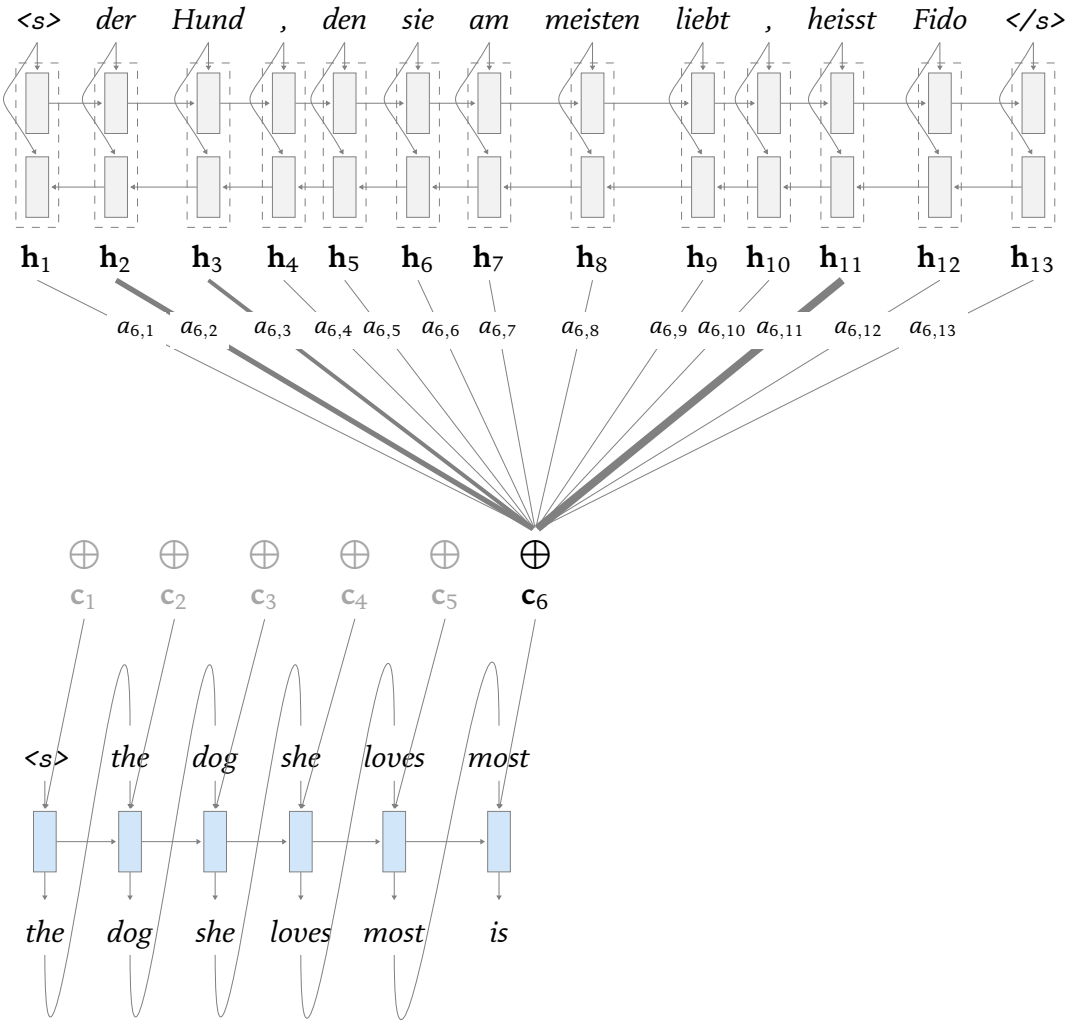


Figure 2.8: Neural sequence to sequence model with attention (example).

et al., 2014). Combined with subword modelling (Sennrich et al., 2016b), it was the first to significantly outperform the best phrase-based systems in multiple languages at the Conference of Machine Translation's (WMT) news translation task (Bojar et al., 2016a; Sennrich et al., 2016a). The model was implemented in a number of open source frameworks for MT (e.g., Sennrich et al., 2017; Junczys-Dowmunt et al., 2018), which accelerated the transition to neural MT in both academia and the language services industry (e.g., Levin et al., 2017).

The Transformer Model

At the time of writing, the best-performing neural MT systems are based on the Transformer model (Barrault et al., 2019). Introduced by Vaswani et al. (2017), it replaces the recurrent components in previous architectures (Sutskever et al., 2014; Bahdanau et al., 2015) with multi-head self-attention.

A multi-head self-attention layer computes a weighted sum of its inputs. The Transformer model uses scaled dot-product instead of what Vaswani et al. (2017) refer to as additive attention (Equation 2.24) due to better time and space efficiency. The layer's output is concatenated from the output of multiple attention functions, referred to as attention heads. Vaswani et al. (2017) find evidence that multiple heads learn to focus on different aspects – in some cases relating to syntactic and semantic features – of the input.

The encoder and decoder consist of a stack of identical encoder and decoder layers, respectively. Each encoder layer has two sub-layers: a multi-head self-attention layer, followed by a feedforward network. In each decoder layer, the first sub-layer attends to the previously generated output words, the second attends to the output of the encoder (representing the input words), and the third is a feedforward network. Residual connections are placed around each sub-layer in the encoder and decoder, and the final output is normalised through layer normalisation (Ba et al., 2016). The architecture is depicted in Figure 2.9.

Besides better translation performance, the removal of recurrent layers allows for faster training because of the amount of computation that can be parallelised. Positional embeddings, added to the input word embeddings, ensure that the model can still make use of their position within the source sentence.

2.2.3 Evaluation

Just as with translations produced by humans, the evaluation of MT is difficult because there is no single correct translation for any given source text (Section 2.1.4). Nevertheless, methods to benchmark MT systems are a necessity. Researchers and developers need quality metrics to make informed choices on model parameters and data sets while building a method or system. MT users, on the other hand, need an indication of quality to

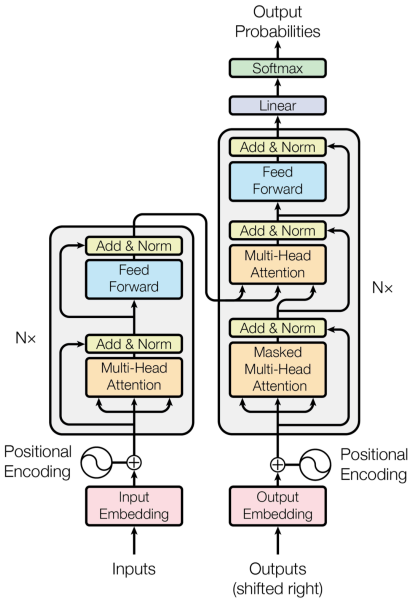


Figure 2.9: The Transformer architecture as illustrated by Vaswani et al. (2017).

chose the best system available for gisting, post-editing (Section 2.1.3), etc., or to decide if MT is viable for their use case at all.

MT systems can be evaluated for various aspects such as quality, productivity, or usability.[11] In this section, we introduce common methods for evaluating the quality of outputs from MT systems. Methods for evaluating productivity and usability will be discussed in the context of mixed-initiative translation (Section 2.3.3).

Koehn (2010, p. 220f) lists four desiderata for MT quality evaluation:

- low cost, i.e., the evaluation process should be quick and resource-saving;
- meaningfulness, i.e., the result of the evaluation should be easy to interpret;
- consistency, i.e., repeated evaluations should lead to the same result; and
- correctness, i.e., the result should be truthful.

The correctness criterion demands that good system outputs will lead to a good evaluation result. Since there is no ground truth for 'good' in translation, some MT evaluation methods use one or multiple human reference translations as a proxy.

[11] These aspects are often, but not necessarily, interleaved. In the context of MT post-editing, a common misconception is that increased quality will lead to higher productivity. However, indicators for what is commonly considered higher quality – such as better word choice and grammar – may not impact post-editing productivity as much as (or at all compared to) other features in MT systems or users, such as correct handling of whitespace and markup ('tags') or training, experience and perceptions, respectively (e.g., Parra Escartín and Arcedillo, 2015).

Resources

The evaluation of an MT system presupposes test data, a method, and one or more agents who carry out the evaluation. In the case of human evaluation, the method is a set of instructions or guidelines, and the agents are human subjects. In the case of automatic evaluation, the method is a well-defined algorithm, and the agent is a machine.

In terms of data, it is good practice to deduplicate and then split the translations available to train an MT system into a training, a development, and a test set. The test set is not used to optimise the MT system because, typically, the aim of an evaluation is to assess how good the MT system is at translating unseen text. If the original set of translations is not deduplicated or if translations from other sources are used for testing, it is important to control for overlap with the training and development set.

Details on human and automatic evaluation methods are given in the sections below. The former will be relevant in Chapter 3, where we study the impact of linguistic context on human evaluation of MT. The latter and the BLEU metric in particular (Page 32) will be used to benchmark a number of MT methods that incorporate prior knowledge, such as partial translations provided by a user, in Chapter 6.

Human Evaluation

The quality of MT output may be evaluated by human judges, henceforth raters. In general, human evaluation is considered more meaningful and correct, but less cost-effective and consistent than automatic evaluation.

Various methods have been proposed for the human evaluation of MT quality.[12] What they have in common is that the MT output to be rated is typically paired with a translation hint – the source text and/or a reference translation – and the raters either adapt (e.g., in the case of HTER, see Snover et al., 2006) or rate the MT output with reference to the translation hint(s). The latter is more common (Bojar et al., 2016b).

In order to minimise costs, the people to serve as raters are often volunteers – such as MT researchers (e.g., Barrault et al., 2019) – or crowd workers (Callison-Burch, 2009). Furthermore, because a sufficient number of ratings is needed to test for statistical significance when comparing two or more systems, raters do not rate a small number of translated texts, but a large number of translated sentences. The sentences are presented in random order, and the origin of any translation – i.e., the name and features of the MT system that produced it – are not shown to raters. Sentence-based evaluation is not problematic as long as the MT systems under evaluation translate sentences independently of each other which, until recently, was the case with state-of-the-art systems (Junczys-Dowmunt, 2019). When sentence-level MT systems are benchmarked against document-level MT systems or human translators, however, the evaluation of isolated sentences disadvantages the latter, as we show in Chapter 3.

[12] For a more extensive overview, see Castilho et al. (2018a).

5	All meaning
4	Most meaning
3	Much meaning
2	Little meaning
1	None

(a) Adequacy

5	Flawless English
4	Good English
3	Non-native English
2	Disfluent English
1	Incomprehensible

(b) Fluency

Table 2.1: Absolute scales for MT quality evaluation (Koehn and Monz, 2006).

MT quality is typically measured along two dimensions: adequacy and fluency. Consider the following example:

> *Source (German)*
> Inulin zum Beispiel hilft beim Aufbau einer gesunden Darmflora.
>
> *MT System 1 (English)*
> Inulin, for example, helps build healthy intestinal flora.
>
> *MT System 2 (English)*
> Inulin, for example, helps to build a healthy intestinal flora.

The evaluation of adequacy requires knowledge of both the source and target language, i.e., bilingual raters.[13] In the human evaluation campaign at WMT 2019, adequacy was defined as the degree to which the target text (MT) expresses the meaning of the source text (Barrault et al., 2019, p. 16). The evaluation of fluency only requires knowledge of the target language, i.e., monolingual raters. Graham et al. (2013) used the following instruction to assess fluency: 'Read the text below and rate it by how much you agree that: The text is fluent English.'

Absolute Scales Traditionally, adequacy and fluency have been rated on 5-point adjectival scales as shown in Table 2.1. The scores obtained can be averaged by system to make conclusions such as 'MT system 1 and 2 scored 3.48 and 4.17 in terms of adequacy, respectively', and/or used to assess statistical significance between any two systems under evaluation (see Koehn and Monz, 2006). To account for between-rater variance – i.e., the fact that some raters will assign generally lower or higher scores than others – per-rater means can be normalised (Koehn, 2010, p. 219), which leads to more consistent outcomes (Graham et al., 2013).

Traditional 5-point adjectival scale judgements are considered problematic in terms of consistency. Since the categories are hard to distinguish, repeated evaluation of the same

[13] Adequacy can be rated by monolingual raters by means of reference-based evaluation, where the system output is compared to a reference translation rather than the source text. However, this form of evaluation has been shown to bias raters (Fomicheva and Specia, 2016).

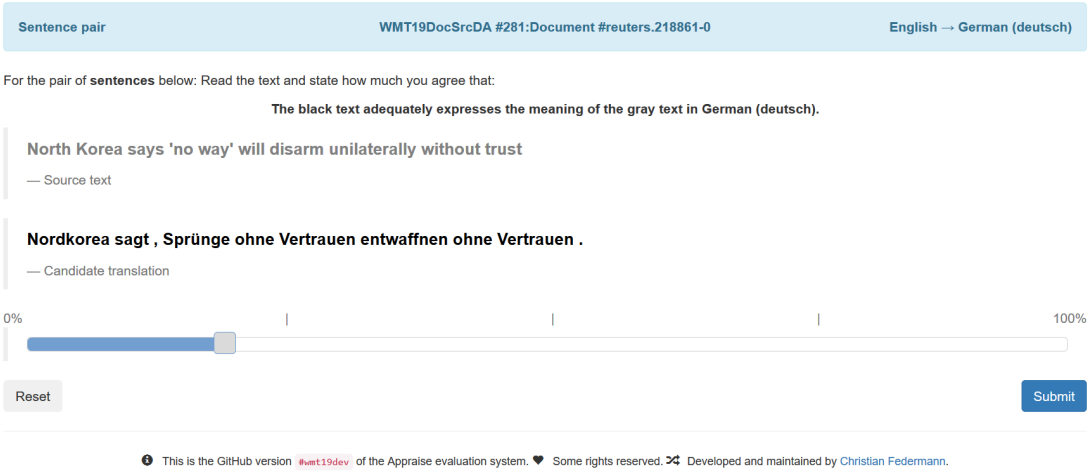


Figure 2.10: Source-based Direct Assessment at WMT 2019 (Barrault et al., 2019).

MT outputs can lead to different results. Consider the output of MT System 1 in the example on Page 29. The sentence

> Inulin, for example, helps build healthy intestinal flora.

is certainly not Flawless English (5), but whether it is Good (4), Non-native (3), or Disfluent English (2) is hard to decide, and different raters – or even the same rater if asked twice – may arrive at different conclusions.

Referred to as Direct Assessment (DA), Graham et al. (2013) circumvent this problem by using a continuous Likert-like scale, presented to raters as a slider (Figure 2.10). In placing the slider between 0 (strongly disagree) and 100 (strongly agree), raters express how much they agree that a translation adequately expresses the meaning of its source text (adequacy) or is fluent English (fluency). Paired with techniques for quality control (see Graham et al., 2013), DA with volunteers and crowd workers has been the primary human evaluation method used at WMT since 2017 (Barrault et al., 2019, p. 16).

Relative Scales Nevertheless, scoring translations on absolute scales remains difficult, and '[r]eplacing this with an ranked evaluation [sic] seems to be more suitable.' (Koehn and Monz, 2006). The motivation for using relative scales in MT quality evaluation is that ranking two (or more) translations of the same source text, produced by different MT systems, seems easier than assigning an absolute score to each of them. Consider the example on Page 29: in terms of fluency, the translation produced by MT System 2 is clearly better than the translation produced by MT System 1. Saying how much better, in contrast, is more difficult: the translation produced by MT System 2 may be eligible for a score of 100/100, but an adequate score for the translation produced by MT System 1 is hard to choose (75/100? 81/100?).

The instruction for raters in relative ranking tasks is typically phrased as follows: 'You are shown a source sentence followed by several candidate translations. Your task is to rank

the translations from best to worst (ties are allowed)' (Bojar et al., 2013). If exactly two systems are evaluated, the outcome of a relative ranking campaign is easy to interpret, e.g., MT System 1 was better than MT System 2 in 41 cases, worse in 59 cases, and the same (tie) in 12 cases. Statistical significance can be tested with a two-tailed sign test.[14] If three or more systems are evaluated, the number of wins and losses are calculated for every pair of systems involved (e.g., 1–2, 2–3, and 1–3), and the TrueSkill method (Sakaguchi et al., 2014) can be employed to assign systems to ranks (clusters) denoting statistically significant differences.

Relative ranking has been shown to improve consistency (i.e., better inter- and intra-rater agreement) in MT evaluation campaigns (Callison-Burch et al., 2007) compared to absolute scoring, but only the latter allows for conclusions to be drawn about the order of magnitude of any differences (i.e., *how much* better one system was than another). At the time of writing, both approaches are used in practice. We will revisit them in Chapter 3, focussing on problems that arise when benchmarking MT systems against human translators.

Automatic Evaluation

Although less meaningful and correct than human quality evaluation, automatic evaluation metrics are indispensable in MT research and development because they provide instant and reproducible results.

The basic idea of automatic quality evaluation is comparing an MT system's output h, referred to as hypothesis, to a reference translation r. An automatic evaluation metric is a function σ that defines the similarity between h and r:

$$\text{score} = \sigma(h, r) \tag{2.25}$$

Scores typically range between 0.0 and 1.0, and are reported in percent.

Traditional NLP evaluation metrics are not applicable because they allow for too much or too little variation between h and r. Precision, the percentage of h words contained in r, is too permissive because word order is irrelevant: for r = 'inulin helps to build a healthy intestinal flora' and h = 'a build flora healthy helps intestinal inulin to', it is 100 % (best). Word error rate (WER), the word-based minimum edit distance between h and r divided by the number of words in r, is too restrictive because any variation is penalised: it is 100 % (worst)[15] for the aforementioned reference r and a perfectly valid hypothesis

[14] Ties are typically ignored (e.g., Bojar et al., 2013), such that the number of successes x is the number of ratings in favour of MT System 1 (here: 41), and the number of trials n is the number of all ratings except ties (here: 100). Ignoring ties may be suboptimal if they are numerous, i.e., the systems are considered very similar. Emerson and Simon (1979) suggest adding half of the ties to x and the total number of ties to n in such cases.

[15] By definition, WER will exceed 100 % if the number of edits exceeds the number of reference words, so an error rate of 100 % is not, strictly speaking, the worst possible score.

with a different wording, such as h = 'the building of a healthy intestinal flora is supported by inulin'.

The aim – and the fundamental problem – of automatic evaluation metrics is to allow variation that is linguistically valid, and penalise variation that is not. In terms of word order, a common approach is to consider multi-word sequences (N-grams) in addition to single words (i.e., unigrams). TER (Snover et al., 2006), for example, extends WER in that the movement of an N-gram of any length is counted as a single edit operation.

However, other forms of valid linguistic variation cannot be inferred from a single reference translation. Consider the case of symonymy: the hypothesis 'for instance' would be a perfectly valid alternative to the reference 'for example' in many contexts, but this is impossible to model with a function that merely compares the two strings without additional resources. To mitigate this problem, most automatic metrics are designed to consider multiple reference translations of the same source text, the hope being that reference translations from different creators (human translators) will exhibit some degree of valid variation. Unfortunately, MT is predominantly evaluated with a single reference per source text because obtaining multiple reference translations is too expensive, and even a set of three or five reference translations will not be exhaustive because, theoretically, the number of valid translations for any source text is unbounded (Section 2.1.4). Some automatic metrics thus use additional resources. METEOR, for example, rewards matches between r and h if they have the same stem or belong to the same semantic class (Banerjee and Lavie, 2005).

We mostly use BLEU (Papineni et al., 2002) for automatic evaluation in this thesis. BLEU measures the N-gram precision P between r and h. It does not consider recall, but penalise short hypotheses with a brevity penalty BP. The metric is defined as

$$\text{BLEU} = BP \cdot P = \min\left(1.0, \exp\left(1 - \frac{\text{length}(r)}{\text{length}(h)}\right)\right) \cdot \left(\prod_{n=1}^{N} \lambda_n p_n\right)^{\frac{1}{N}}, \tag{2.26}$$

where n is the N-gram precision of order n, N is the highest N-gram order, and λ_n is the weight of the N-gram precision of order n. N is typically set to 4, and λ_n to 1.0 for every n (i.e., uniform weights). With multiple references, overlaps with any reference are counted in P, and the length of the reference that comes closest to the length of the hypothesis h is used in BP.

While multi-reference BLEU was the first metric shown to correlate with human judgements of MT quality (Papineni et al., 2002), there is growing evidence that the metric becomes less reliable as MT quality improves.[16] Callison-Burch et al. (2006) have criticised BLEU for not considering the fact that some words (e.g., content words) are more important than others (e.g., function words), and warned against comparing BLEU scores from different evaluation settings (such as language pairs and test sets).

[16] At WMT 2019, human quality judgements for the strongest MT systems were negatively correlated with BLEU (Ma et al., 2019, p. 79).

2.3 Mixed-initiative Translation

The emergence and growing adoption of translation technology has long turned professional translation into a form of human–computer interaction (HCI): a mixed-initiative task (Carbonell, 1970) in which precision-oriented humans and recall-oriented machine agents take turns translating a text. In this section, we introduce the concept and briefly summarise the historical trajectory of interactive MT (IMT) (Section 2.3.1). We then turn to actual implementations of this concept, describing what modern software workbenches for mixed-initiative translation typically entail (Section 2.3.2), and how they can be evaluated (Section 2.3.3).

2.3.1 Interactive Machine Translation

The idea of combining the strengths of MT systems and professional translators goes back to at least the 1950s. In 1951, Yehoshua Bar-Hillel published a comprehensive report on research activities in MT, concluding that research had concentrated too much on fully automatic high-quality MT (known as FAHQMT), which seemed unrealistic to him in the near future. One of his key arguments was that no method was feasible by which machines would accurately resolve semantic ambiguities. Consequently, he argued for 'mixed MT, i.e., a translation process in which a human brain intervenes' in the form of pre- and post-editing (Bar-Hillel, 1951). However, the MT research community kept focusing on fully automatic approaches (Section 2.2.1), which eventually lead to widespread disillusionment and the cutting of funds when the 1966 ALPAC report found that more than a decade of research had produced systems of little practical value (Pierce et al., 1966). Their output did indeed require human post-editing, which 'took slightly longer to do and was more expensive than conventional human translation' (ibid., p. 19). This evidenced an interface problem: while translations produced by early MT systems contained valid portions, human users were unable to leverage them so as to increase their productivity. In this light, MT becomes an HCI problem: how best to present machine suggestions to human users?

The key aspect of this question is who does what and when. The first IMT systems, which were theorised but never actually implemented, followed Licklider's (1960) idea of a 'symbiotic partnership' in which humans would set goals and evaluate results, while machines would do the 'routinizable work'. In Bisbey and Kay's (1972) MIND system, for example, monolingual 'consultants' would help disambiguate source texts (pre-editing) before the machine took over and transferred them into the target language, where monolingual 'editors' would then ensure target language fluency (post-editing). The problem of this serial process is that translators cannot influence decisions *while* a text is being translated. If an MT system chooses a wrong translation for an ambiguous word at the beginning of a sentence, for example, the translation of the subsequent words will be conditioned on that wrong choice (Equation 2.18) and thus likely be inadequate, too; the translator will have no choice but to post-edit a large portion of the

sentence. The idea of IMT is to turn this serial into a cyclic process: the system offers a set of recommendations, the translator chooses the most suitable recommendation, the system generates the next set of recommendations based on that choice, etc., turning the translator's role into avoiding rather than correcting mistakes.

Church and Hovy (1993), who described post-editing as an 'extremely boring, tedious and unrewarding chore', suggested to realise this cyclic process by means of autocompletion: 'the [translator's] workstation could have a "complete" key ... which would fill in the rest of a partially typed word/phrase from context.' Terming the approach as target-text mediated IMT,[17] Foster et al. (1997) laid out the technical foundation. The probability of a target text word y_t is conditioned on its preceding, user-defined words $y' = y_1, \ldots, y_{t-1}$ and the source text $x = x_1, \ldots, x_s$:

$$P(y_t|y', x) \tag{2.27}$$

Foster et al. model this distribution as a linear combination of separate predictions from a language model $P(y_t|y')$ and a translation model $P(y_t|x)$,

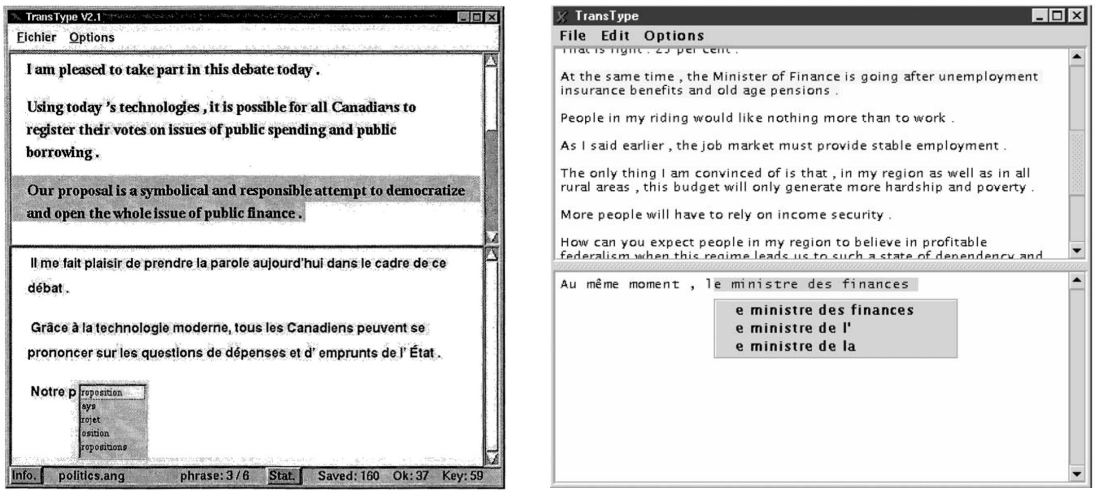
$$P(y_t|y', x) = \lambda\, P(y_t|y') + (1 - \lambda)\, P(y_t|x). \tag{2.28}$$

The interpolation coefficient λ was meant as a context-sensitive switch (i.e., $\lambda \in [0, 1]$) to use either the language or the translation model when predicting a word in a particular context, triggered by a feature function $\phi(y', x)$. Foster et al. (1997) experimented with several target text features such as the previous bigram, the frequency of the previous bigram in the training data, the current part-of-speech, or the sentence position, but eventually opted for a context-independent scalar coefficient optimised on held-out development data (e.g., λ=0.6 in Langlais and Lapalme, 2002) because results were discouraging. The choice of a linear combination over a noisy channel decomposition of Equation 2.27[18] was motivated by speed. While making a stronger independence assumption, the linear combination is less expensive to compute, and 'speed is crucial' in IMT because 'the system will need to generate a new translation of the source text for each character a translator types, fast enough so as never to force even the swiftest typist to wait' (Foster et al., 1997).

TransType (Langlais et al., 2000), the first evaluated IMT system, used Foster et al.'s (1997) method to suggest completions for target text words that users had started typing. A generator component would select all words that overlapped with the user's input from the system's vocabulary, score them (Equation 2.28), and display the most probable words in a drop-down widget (Figure 2.11a). Later versions provided phrase-level suggestions (Langlais and Lapalme, 2002, Figure 2.11b) and sentence completion (Esteban et al., 2004; Barrachina et al., 2009).

[17] Because translators would not focus on disambiguating source texts that go into the MT system, but continuously evaluate translation suggestions that it generates as they are writing the target text.

[18] i.e., $P(y_t|y', x) \propto P(x|y_t, y')P(y_t|y')$, see Equation 2.3



(a) Word-level
(Langlais et al., 2000)

(b) Phrase-level
(Langlais and Lapalme, 2002)

Figure 2.11: Target text suggestions in TransType.

Knowles and Koehn (2016) replaced the statistical models with a recurrent encoder-decoder model with attention (Section 2.2.2). Recall that with this model, the decoding process involves computing a probability distribution over all words in the target language vocabulary at each timestep, selecting the most (greedy) or the k most likely words (beam search), and feeding the selection into the next timestep, where the process repeats (Figure 2.5 shows an illustration in the context of language modelling). Rather than the most likely word $\hat{y}$, we can enforce the next word in y' at each timestep until the prefix is fully consumed, and only then proceed with regular prediction – until a full word or the end-of-sequence symbol is generated for word-level suggestions or sentence completion, respectively. To this end, Equation 2.22 can be rewritten as

$$p(y_t|y', x) = g(y', \mathbf{s}_t, \mathbf{c}_t). \tag{2.29}$$

In addition to better prediction accuracy over the statistical approach (Knowles and Koehn, 2016), an advantage of this method is that the encoder-decoder model can be trained as usual, and then used for both regular and prefix-constrained MT.

A major limitation of the IMT approaches discussed here is that translation suggestions can only be generated in a left-to-right fashion. If a translator wishes to get a suggestion for the beginning of a sentence given a specific ending, or get a suggestion for an entire sentence given a number of translated words or phrases, the decoding process needs to be reorganised. We suggest an alternative approach, and discuss a number of related approaches, in Chapter 6.

IMT functionality is not (yet) widespread in commercial software workbenches for professional translation, which are introduced in the next section. Some workbenches of-

fer prefix-based word- and phrase-level suggestions,[19] but to the best of our knowledge, no commercial workbench offers sentence completions that are continuously updated as translators make adjustments.[20]

2.3.2 Software Workbenches

Software workbenches for professional translation, commonly called CAT tools,[21] combine a bilingual text editor (the frontend) with machine agents that provide translation suggestions (a number of backend systems).

CAT tools abstract form from content. They extract text to be translated from various document formats (Figure 2.12), and reinsert text into the original document format after translation. This serves two purposes: user experience and productivity.

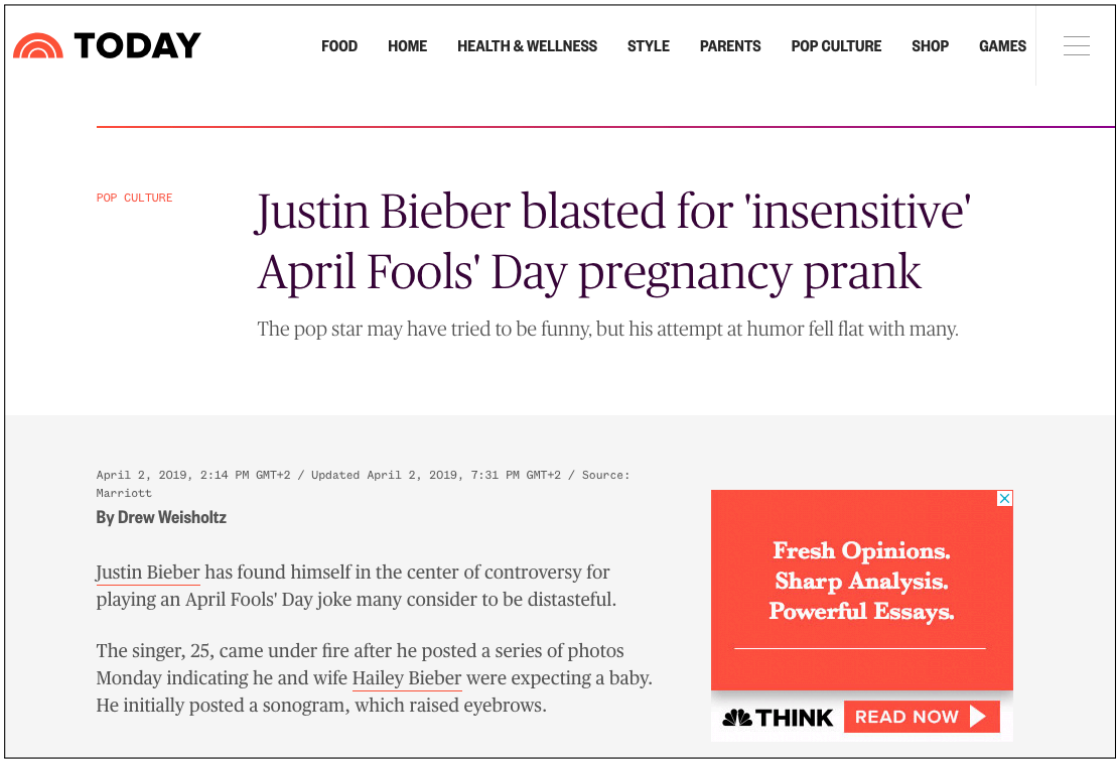
The abstraction is meant to help translators focus on translatable text. The text of a website, for example, can be translated without worrying (or even knowing) about HTML syntax and conventions. The abstraction consists of two subsequent processes: filtering, in which translatable text is identified and extracted from the document's source code; and segmentation, in which the extracted text is split into translation units, such as paragraphs or sentences. These processes do not require human intervention. They run in the background when a user imports a document into the CAT tool, before they start translating. An example is shown in Figure 2.13. The abstraction allows for a user-defined visualisation of translatable text, such as large font size and black colour regardless of font size(s) and colour(s) in the original document.

The second purpose, and the main driver for industry adoption (Somers, 2003), is productivity. The normalisation of text enables users to re-use translations in other contexts. If they translate the segment 'Pop Culture' as 'Popkultur' in one document, the CAT tool will store this translation in a TM and suggest it to the user when they need to translate 'Pop Culture' in another document, regardless of it appearing in bold red font in the former and as a grey bullet point in the latter. CAT tools will not only suggest exact, but also fuzzy matches (Section 2.1.1). Matches with scores above a certain threshold are shown to the user, who may choose the most suitable and adapt its non-conforming parts instead of translating from scratch. Segments can be configured to be sentences or paragraphs in CAT tools, but since it is more likely for a single sentence to occur in another document than an entire paragraph, sentence segmentation promises higher yield (i.e., more exact and fuzzy matches) and is more common in practice.

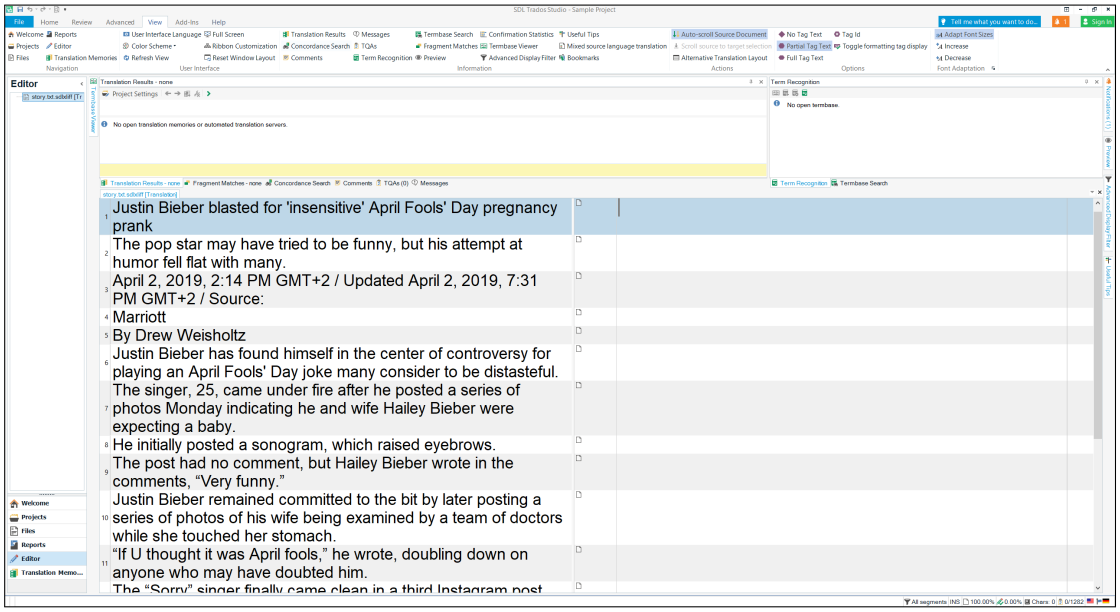
[19] These suggestions can also stem from TMs and TBs, not only MT.

[20] Lilt offered continuously updating prefix-based sentence completion in the form of ghost text (see Green et al., 2014a), but the workbench is no longer available to the general public.

[21] CAT is for computer-aided translation. CAT tools are also referred to as Translation Memory Systems (e.g., Somers, 2003), but since modern software workbenches for professional translation combine multiple backend systems to provide translation suggestions, we use TM to refer to the backend system that provides stored segments, and CAT tool to refer to the software workbench as a whole.



(a)



(b)

Figure 2.12: A website (a) as shown in a CAT Tool (b).

```
<p class="endmarkEnabled">The
singer, 25, came under fire
after he posted a series of
photos Monday indicating he and
 wife <a href="https://www.
today.com/popculture/justin-
bieber-hailey-baldwin-get-real-
about-love-marriage-vogue-
t148296" title="Show␣related␣
article" target="_blank">Hailey
 Bieber</a> were expecting a
baby. He initially posted a
sonogram, which raised eyebrows
.</p>
```

(a)

The singer, 25, came under fire after he posted a series of photos Monday indicating he and wife {1}Hailey Bieber{2} were expecting a baby.
Show related article
He initially posted a sonogram, which raised eyebrows.

(b)

Figure 2.13: Excerpt from the source code of a HTML document (a) as filtered and segmented by a CAT Tool (b).

Translation suggestions from TMs are increasingly combined with translation suggestions from MT systems (Pielmeier and Lommel, 2019). MT is typically included among fuzzy matches in the list of translation suggestions to chose from, or automatically inserted into the target document in a pre-translation step. The display or insertion of MT can be conditioned on the presence of TM matches above a certain threshold (high fuzzy matches) or other factors such as segment length (e.g., more than 40 words), the assumption being that TM matches will require less post-editing than MT in such instances. The difference between the two is not only that MT may contain (more) errors, but also that, because the source side of a fuzzy match can be compared to the new segment to be translated, its matching and non-matching parts (substrings) can be highlighted, and a match score can be calculated (see above).[22] MT can also be used to 'repair' fuzzy matches, i.e., to translate and replace the differing parts rather than translate the entire source segment (Koehn and Senellart, 2010; Bulté and Tezcan, 2019).[23] MT suggestions are typically static: the source segment is translated exactly once, and the suggestion is not updated as the user makes adjustments.

[22] Both are important cues of confidence to translators. In particular, the popularity of segment-level confidence scores among professional translators is highlighted in Moorkens and O'Brien's (2017) survey on user interface needs for post-editing, in which 81 % of 233 respondents expressed that they 'would like to be presented with confidence scores for each target text segment from the MT engine.'

[23] In SDL Trados, for example, this feature is called 'upLIFT Fuzzy Repair' and leverages sub-segment matches from TMs and TBs in addition to MT (Flanagan, 2014).

2.3.3 Evaluation

Software workbenches for professional translation have been evaluated primarily by means of surveys and user studies, with research questions centring around user experience and productivity. Surveys are more common for research on commercial and well-established products (e.g., O'Brien et al., 2017; Schneider et al., 2018), while user studies are often conducted with research prototypes (e.g., Green et al., 2014a; Coppers et al., 2018). In this section, we give a brief overview of the typical research designs and metrics used in CAT Tool evaluations.[24]

User Experience

The International Standard on human-centred design for interactive systems (ISO 9241-210:2010) defines user experience as a 'person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service'. Aspects like how users feel when using a piece of software, how well they understand it, as well as how well it serves their purpose and fits into the context in which they are using it (Alben, 1996) are relevant in our context because ill-conceived software can cause irritation (O'Brien et al., 2017) and resistance (Cadwell et al., 2018) in professional translators even if it promises higher productivity.

Insights about user experience can be gained through observation and interrogation. To observe how translators interact with CAT tools, one option is to implement mechanisms to collect user activity data such as key strokes, mouse movements, or eye movements. This data can be analysed qualitatively or quantitatively. User activity data (or screen recordings) can be used to generate replays of experimental sessions in which participants are given a translation task, and in a qualitative analysis, experts look for patterns in how participants interact with specific features of the CAT tool in these replays. During development of TransType 2, for example, Macklovitch et al. (2005) realised that '[s]ome of the features we worked hard to develop based on our own experience (e.g. using the mouse for entering partial completions or using cut and paste) were almost never used by the professionals', which allowed them to refocus on features that actually mattered to translators (as urged by O'Brien, 2012). A quantitative analysis, on the other hand, typically entails statistical analysis of user activity data, such as analysing how often or how long on average participants use a certain feature. Green et al. (2014a), for instance, found that translators made frequent use of interactive features for target text generation, but almost no use of interactive features for source text comprehension in their IMT system, suggesting that the former are more important when designing CAT tools.

While mechanisms for collection of user activity data have the advantage of being non-intrusive since keyboard and mouse input can be logged without distracting translators as they work on a task, their implementation is laboursome and often inviable with commercial software. An alternative approach to user observation is the conduct of ethno-

[24] For a more comprehensive overview, see Läubli and Green (2019).

graphic studies, in which experts observe and possibly interrogate translators in real-life settings. LeBlanc (2013), for example, spent close to 300 hours at three translation agencies in Canada, conducting interviews and observing translators in half-day shadowing sessions. The study resulted in a comprehensive list of advantages and disadvantages in working with TMs, one of the latter being that sentence segmentation changes the translator's relationship with the text because 'reorganising the [target text] (combining, splitting, moving about sentences) becomes more complicated (if not impossible) and more time-consuming', a finding that motivates our work on document-level interfaces in Chapter 5.

While ethnographic studies combine observation and interrogation, other forms of user experience evaluation focus exclusively on the latter. Cadwell et al. (2016, 2018) collected qualitative data by organising focus group meetings with translators. In transcribing the seeded discussions and coding the transcripts by means of a thematic analysis methodology (Braun and Clarke, 2006), the authors identified several factors for the (non-)adoption of MT among translators, one being 'Quality of CAT environment software' (Cadwell et al., 2018). Feedback on CAT tool usability has also been elicited with written surveys, one advantage being that a larger number of participants (or respondents) can be considered. Perhaps most important with regard to our research questions in Chapters 4 and 5 is the international survey conducted by Ehrensberger-Dow et al. (2016), taken by more than 1,800 professional translators. O'Brien et al. (2017) analysed the survey data with a focus on CAT tool usability, and identified several features that translators find irritating or missing. The issue that caused most irritation was user interface complexity: CAT tools were said to be difficult to navigate and to require too many clicks to complete certain tasks. The second most irritating feature was segmentation: translators reported to have 'problems merging segments' and mentioned 'irritation caused by segmented view of text'.

Productivity

Translation speed is a decisive factor in professional translation due to its direct economic impact, but gains in speed are only meaningful if they are not offset by lower quality. As a result, productivity assessments typically test the impact of a translation condition – e.g., translating with and without the translation aid of interest – on how fast a number of translators, the experimental subjects, can produce translations, and on how good these translations are. The translation aid is said to increase productivity if it allows faster translation at the same or higher quality.

The central design decisions and main challenges in user studies on translation productivity are how to measure speed, how to measure quality, and how to ensure that the results are generalisable. Recording the time it takes subjects to translate a text is straightforward, but because the same subject cannot be presented with the same text in different experimental conditions (see below) and because some experimental designs use a fixed period of time in which subjects are to translate as much text as possible, time recordings must

be normalised to allow for a meaningful comparison between experimental conditions. To give just a few examples, translation speed has been reported in words per hour (Plitt and Masselot, 2010), seconds per word (Koehn, 2009), or seconds per sentence[25] (Green et al., 2014a, log-transformed for statistical analysis). The choice of different normalisations makes it difficult to compare time measurements from different studies, particularly if authors do not report descriptive statistics on texts (such as average sentence length) or do not define what constitutes a word (5 target characters, excluding whitespace? Actual words, including numbers and punctuation?).

The problem of measuring translation quality has been thoroughly discussed in Sections 2.1.4 and 2.2.3. Both automatic (e.g., Green et al., 2014a) and manual evaluation methods have been used in productivity studies, with the latter involving either external experts (e.g., Läubli et al., 2013), crowd workers (e.g., Green et al., 2013), or students (Koehn, 2009). A number of assessments in the context of commercial organisations have also used their internal resources and procedures, such as the seminal study conducted by Plitt and Masselot (2010) at Autodesk, where the 'linguistic quality assurance team reviewed part of the … translation and post-editing jobs for each language' and concluded that 'all would have been published as is'.

Probably the biggest challenge in designing and analysing data resulting from experiments on translation productivity is ensuring that the findings generalise to other settings. While an in-depth discussion is beyond the scope of this section, and some further details will be discussed in Chapter 5, a major problem is that many studies are conducted with a small number of subjects (e.g., six in Läubli et al., 2013), a single language pair (e.g., English to Dutch in Coppers et al., 2018), and/or a single type or genre of source texts (e.g., medical package leaflets in Alabau et al., 2015). The limited availability of (or budget for) qualified subjects is one of the reasons why within-subject designs, where all subjects are exposed to all experimental conditions, are predominant in productivity studies. To the best of our knowledge, the largest study on computer-assisted translation so far has been conducted by Green et al. (2014a), who involved 16 professional French to English and 16 professional English to German translators in a full-day experiment.

[25] Other studies have used similar units. Etchegoyhen et al. (2014), for example, measured translation speed in subtitles per minute.

Chapter 3

Document-level Evaluation of Translation Quality

MT has made astounding progress in recent years thanks to improvements in neural modelling (Section 2.2.2), and the resulting increase in translation quality is creating new challenges for MT evaluation (Section 2.2.3). Some recent results suggest that neural MT 'approaches the accuracy achieved by average bilingual human translators [on some test sets]' (Wu et al., 2016), or even that its 'translation quality is at human parity when compared to professional human translators' (Hassan et al., 2018).

Claims of human parity in MT are certainly extraordinary, and require extraordinary evidence. In this chapter,[1] we reassess Hassan et al.'s (2018) evaluation, showing that the finding of parity between their strongest MT system and professional human translation (HT) in Chinese to English news translation is owed to weaknesses in the evaluation design: when professional translators (rather than crowd workers) evaluate full news articles (instead of randomly drawn sentences), HT scores significantly higher in terms of both accuracy and fluency. Our empirical findings (Section 3.4) and error analysis (Section 3.5) make a strong case for revisiting best practices in MT evaluation. However, an evaluation of full documents is challenging in practice because large sample sizes are needed for sufficient statistical power, and rating hundreds of documents rather than hundreds of sentences is considerably (if not prohibitively) more expensive. To that end, we compare and contrast our results with further findings of 'human parity' and 'super-human performance' in MT evaluations, and review a number of alternative evaluation protocols (Section 3.6). We conclude the chapter with a set of recommendations for assessing strong MT systems in general, and human–machine parity in particular (Section 3.7).

[1] The main findings presented in this chapter have been published as a conference paper (Läubli et al., 2018b) and synthesised with concurrent work in a journal article (Läubli et al., 2020a). The presentation herein includes further examples of differences between HT and MT, and depicts the evaluation materials that were shown to raters.

3.1 Background

Hassan et al. (2018) train several systems for automatic Chinese to English translation, extending the Transformer model (Page 26) with mechanisms aimed at better exploitation of bi- and monolingual training data (Dual Learning and Joint Training) and better recovery from suboptimal word choices during output generation (two-pass decoding with Deliberation Networks and Agreement Regularisation). The systems are trained with a subset of the WMT 2017 Chinese–English training data obtained by means of data selection and filtering. Hassan et al. combine these systems through n-best list reranking in several configurations, and assess the three system combinations achieving the best BLEU scores on the WMT 2017 Chinese–English test set – Combo-4, Combo-5, and Combo-6 – in a human evaluation campaign.[2]

The evaluation campaign follows best practices in MT evaluation (Section 2.2.3). Bilingual raters score translated sentences in a source-based Direct Assessment (DA, Figure 2.10). The source sentences for the DA tasks are sampled from the WMT 2017 Chinese–English test set, and the translations stem from the aforementioned system combinations as well as six additional sources:

M_1: MT from Microsoft's production system (October 2017).

M_2: MT from Google's production system (October 2017).

M_3: MT from the best MT system for Chinese to English news translation at WMT 2017 (Wang et al., 2017).

H_A: HT from the WMT 2017 Chinese–English test set, i.e., the official reference translations.

H_B: HT ordered from a translation agency, created from scratch (i.e., without using any MT). These translations were ordered since H_A was found to contain errors.[3]

H_C: Post-edits of M_2, ordered from a translation agency.

Hassan et al. have the entire test set (2001 sentences) translated by each of these nine translation sources, and collect scores from three raters for each translated sentence. A Wilcoxon rank sum test at $p \leq .05$ shows no significant difference between the normalised scores for Combo4–6 and H_B, through which the authors argue that the outputs of their 'research systems are indistinguishable from human translations' (Hassan et al., 2018, p. 15).

[2] For references and implementation details, see Hassan et al., 2018, Sections 3.3–3.6 and 4.1.

[3] At WMT 2018, the organisers themselves noted that 'the manual evaluation included several reports of ill-formed reference translations' (Bojar et al., 2018, p. 292).

3.2 Hypothesis

Laudably, Hassan et al. have released their data publicly to allow for external validation of their claims.[4] Our main interest lies in the evaluation protocol, and we empirically investigate if the lack of document-level context could explain the inability of human raters to find a quality difference between human and machine translations. We test the following hypothesis:

> A professional translator who is asked to rank the quality of two candidate translations on the document level will prefer a professional human translation over a machine translation.

Note that our hypothesis is slightly different from that tested by Hassan et al. (2018), which could be phrased as follows:

> A bilingual crowd worker who is asked to directly assess the quality of candidate translations on the sentence level will prefer a professional human translation over a machine translation.

As such, our evaluation is not a direct replication of that by Hassan et al., and a failure to reproduce their findings does not imply an error on either our or their part. Rather, we hope to indirectly assess the accuracy of different evaluation protocols. Our underlying assumption is that professional human translation is still superior to MT, but that these quality differences are not revealed in a sentence-level evaluation with crowd workers.

3.3 Experimental Methods

We conduct an independent evaluation of the professional human translations (H_B) and the outputs of the best MT system (COMBO-6) that were found to be of equal quality by Hassan et al. (2018). We use a 2×2 mixed factorial design, testing the effect of source text availability (adequacy, fluency) and experimental unit (sentence, document) on quality ratings by professional translators.

3.3.1 Task

We elicit quality ratings through pairwise ranking. Subjects are shown two translations of a source text, and are asked which is better (with ties allowed). The source text of the translations is shown in the adequacy condition, but not in the fluency condition. The source texts and translations are single sentences and full documents in the sentence and fluency conditions, respectively. Examples are shown in Figure 3.1.

[4] http://aka.ms/Translator-HumanParityData

3.3.2 Materials

The WMT 2017 Chinese–English test set[5] contains 169 news articles (documents) from various news websites. 123 of these documents are original Chinese; 46 documents are Chinese translations of English documents. Although common practice in MT evaluation, evaluating translations of translations (referred to as 'translationese') rather than original texts has a confounding effect (as discussed in Section 3.6), so we only consider the documents which are original Chinese. We randomly sample 55 documents and 2×120 sentences from these documents[6] to serve as the source texts in our experiment, and pair them with the professional human and machine translations in H_B and Combo-6, respectively.

3.3.3 Subjects

To optimise cost, MT quality is typically assessed by crowd workers or volunteers (Section 2.2.3). Hassan et al. (2018) also obtain quality ratings from crowd workers, but as discussed in Section 3.6, empirical findings by Toral et al. (2018) indicate that crowd workers disregard translation nuances, which leads to a more tolerant judgement of MT systems and lower inter-annotator agreement.

We recruit professional translators from ProZ, a well-known online market place for professional freelance translation.[7] For the adequacy condition, we recruit four Chinese to English translators native in Chinese (two subjects: $S_{1,2}$), English (S_3), or both (S_4); for the fluency condition, we recruit four revisers native in English (S_{4-8}). The subjects have 13.7 years of experience and 8.8 positive client reviews on ProZ on average, and receive USD 188.75 for rating 55 documents and 120 sentences. These averages include an additional reviser (S_9) we recruited when S_8 showed poor performance on document-level spam items (see below) in the fluency condition, whose judgements we exclude from analysis. We also exclude sentence-level results from four subjects ($S_{2,4,6,7}$) because there was overlap with the documents they annotated, which means that we cannot rule out that the sentence-level decisions were informed by access to the full document.

3.3.4 Procedure

Each rater evaluates 55 documents and 120 sentences. To hedge against random ratings, we convert 5 documents and 16 sentences per set into spam items (Kittur et al., 2008): we render one of the two options nonsensical by shuffling its words randomly, except for 10 % at the beginning and end.

Each subject receives one PDF file with 55 documents, and another PDF file with 120 sentences. The order of experimental items as well as the two choices for each item (HT and

[5] `http://data.statmt.org/wmt17/translation-task/test.tgz`, `newstest2017-zhen-src.zh.sgm`
[6] The documents contain 8.13 sentences on average.
[7] `https://www.proz.com`

24 **Article-E-24**

8月11日，首届锦州湿地旅游文化节在锦州东方华地域“鹤泉湖”畔启幕。从2014年起，旅游产业创造的收入，占锦州市GDP10%；2015年更是达到了11.3%。东方华地域湿地温泉旅游区，地处锦州市凌海大有经济区滨海公路凌海段1号，距离锦州市区70公里，距离凌海市区50公里，距离盘锦市70公里。景区有室内温泉占地3000平方米，室外温泉占地4600平方米。

A On August 11, the first Jinzhou Wetland Tourism and Culture Festival kicks off on the banks of Hequan Lake in the eastern Chinese city of Jinzhou. Since 2014, the tourism industry has generated 10% of Jinzhou's GDP, and 11.3% in 2015. Dongfang Huadicheng Wetland Hot Spring Tourism Zone, located in Jinzhou City, Linghai Economic Zone, Binhai Road, Linghai Section 1, 70 kilometers away from Jinzhou City, 50 kilometers away from Linghai City, 70 kilometers away from Panjin City. Scenic area has indoor hot spring covers an area of 3,000 square meters, outdoor hot spring covers an area of 4,600 square meters.

B On August 11, the inaugural Jinzhou Wetlands Tourism and Cultural Festival opened on the banks of "Hequan Lake" at Jinzhou Dongfang Huadi City. Since 2014, the revenue generated by the tourism industry made up 10% of the GDP of Jinzhou City. By 2015, this figure has increased to 11.3%. The wetlands hot spring recreation area at Dongfang Huadi City is located at No. 1, Linghai Section, Binghai Road, Linghai Dayou Economic Area, Jinzhou City, and is only 70 kilometers away from Jinzhou City, 50 kilometers away from Linghai City, and 70 kilometers from Panjin City. The attraction area is provided with 3,000 square meters of indoor hot springs and 4,600 square meters of outdoor hot springs.

(a) Document, Adequacy

24 **Article-E-24**

A On August 11, the first Jinzhou Wetland Tourism and Culture Festival kicks off on the banks of Hequan Lake in the eastern Chinese city of Jinzhou. Since 2014, the tourism industry has generated 10% of Jinzhou's GDP, and 11.3% in 2015. Dongfang Huadicheng Wetland Hot Spring Tourism Zone, located in Jinzhou City, Linghai Economic Zone, Binhai Road, Linghai Section 1, 70 kilometers away from Jinzhou City, 50 kilometers away from Linghai City, 70 kilometers away from Panjin City. Scenic area has indoor hot spring covers an area of 3,000 square meters, outdoor hot spring covers an area of 4,600 square meters.

B On August 11, the inaugural Jinzhou Wetlands Tourism and Cultural Festival opened on the banks of "Hequan Lake" at Jinzhou Dongfang Huadi City. Since 2014, the revenue generated by the tourism industry made up 10% of the GDP of Jinzhou City. By 2015, this figure has increased to 11.3%. The wetlands hot spring recreation area at Dongfang Huadi City is located at No. 1, Linghai Section, Binghai Road, Linghai Dayou Economic Area, Jinzhou City, and is only 70 kilometers away from Jinzhou City, 50 kilometers away from Linghai City, and 70 kilometers from Panjin City. The attraction area is provided with 3,000 square meters of indoor hot springs and 4,600 square meters of outdoor hot springs.

(b) Document, Fluency

104 **Sentence-O-104**

他还说，如果不对极端主义势力进行打击，这些势力会扩展到俄罗斯。

A He added that if extremist organizations are not suppressed, these forces will spread to Russia.

B Without a crackdown on extremist forces, they would expand into Russia, he added.

(c) Sentence, Adequacy

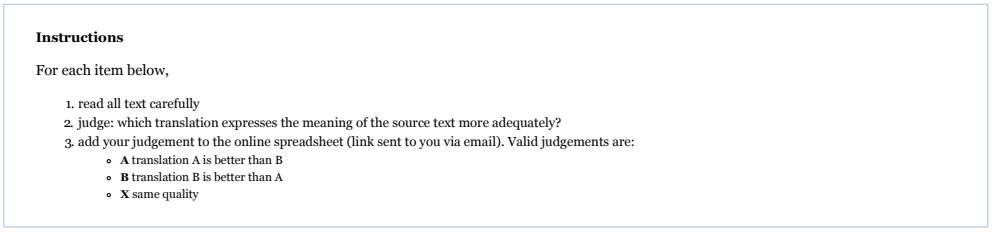
104 **Sentence-O-104**

A He added that if extremist organizations are not suppressed, these forces will spread to Russia.

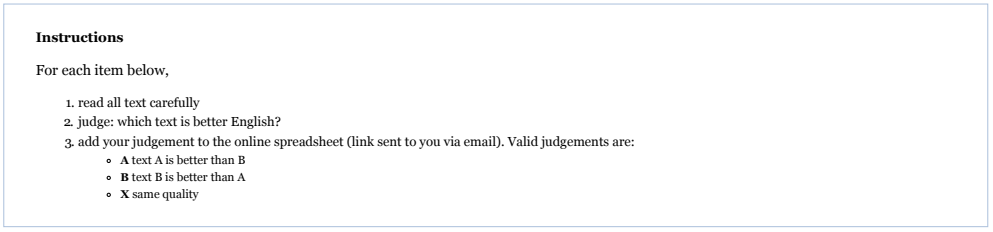
B Without a crackdown on extremist forces, they would expand into Russia, he added.

(d) Sentence, Fluency

Figure 3.1: Examples of experimental items as shown to subjects in the four experimental conditions (a–d).



(a) Adequacy conditions



(b) Fluency conditions

Figure 3.2: Rating instructions as shown to subjects.

MT) is randomised. The rating instructions, placed at the top of each PDF document, are shown in Figure 3.2: we ask subjects to read each text carefully, and record their rating in a designated spreadsheet. The PDF documents and spreadsheets are sent to participants via email, and we ask subjects to complete the assignment within seven days. We remunerate each subject as soon as we receive the spreadsheet with ratings for all experimental items in the assignment.

3.4 Experimental Results

We test for statistically significant preference of HT (H_B) over MT (Combo-6) or vice versa by means of two-sided Sign tests. Let a be the number of ratings in favour of MT, b the number of ratings in favour of HT, and t the number of ties. We report the number of successes x and the number of trials n for each test, such that $x = b$ and $n = a + b$.[8]

In terms of adequacy, MT and HT are not statistically significantly different on the sentence level ($x = 86, n = 189, p = .244$). This is consistent with the results that Hassan et al. (2018) obtained with an alternative evaluation protocol (crowdsourcing and direct assessment, see above). However, when evaluating entire documents, subjects show a statistically significant preference for HT ($x = 104, n = 178, p < .05$). While the number of ties is similar in sentence- and document-level evaluation, preference for MT drops

[8] Emerson and Simon (1979) suggest the inclusion of ties such that $x = b + 0.5t$ and $n = a + b + t$. This modification has no effect on the significance levels reported in this section.

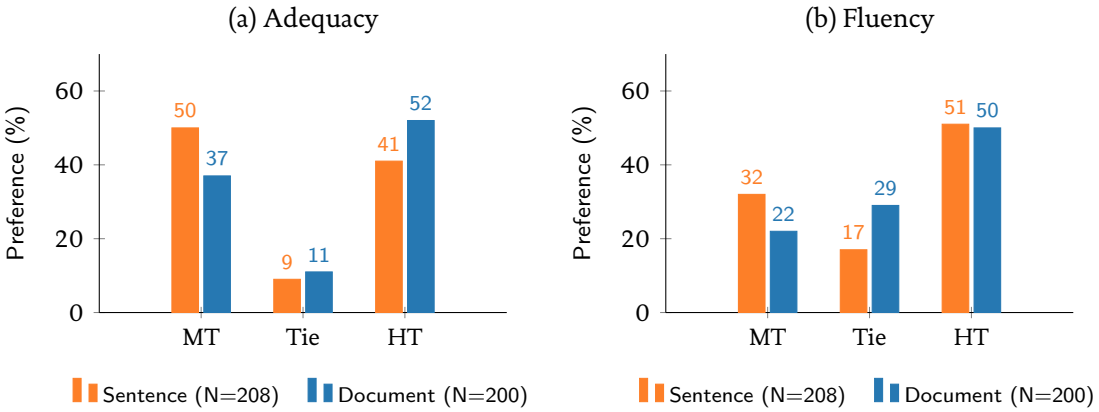


Figure 3.3: Average ratings by experimental condition (in %).

Condition	Document			Sentence		
Aggregation	MT	Tie	HT	MT	Tie	HT
Fluency						
Average	22	29	50	32	17	51
Majority	24	10	66	26	23	51
Adequacy						
Average	37	11	52	50	9	41
Majority	32	18	50	38	32	31

Table 3.1: Aggregation of ratings by experimental condition (in %). Average ratings are visualised in Figure 3.3.

Condition	Document			Sentence		
Subject	MT	Tie	HT	MT	Tie	HT
Adequacy						
S_1	26	0	24	59	3	42
S_2	18	4	28	38	23	43
S_3	10	15	25	44	16	44
S_4	20	3	27	38	11	55
Sum	74	22	104	103	19	86
Fluency						
S_5	13	8	29	30	32	42
S_6	12	14	24	40	14	50
S_7	11	17	22	32	30	42
S_8				36	4	64
S_9	8	18	24			
Sum	44	57	99	66	36	106

Table 3.2: Ratings by subject and experimental condition. Greyed-out fields indicate that raters had access to full documents for which we elicited sentence-level judgements; these are not considered in the results reported in this section (Sum).

Condition	Document	Sentence
Metric		
Fluency		
Same-label	.55	.45
Cohen's κ	.32	.13
Adequacy		
Same-label	.49	.50
Cohen's κ	.13	.14

Table 3.3: Inter-rater agreement by experimental condition.

from 50 to 37 % in the latter (Figure 3.3a).

In terms of fluency, subjects prefer HT on both the sentence ($x = 106, n = 172, p < .01$) and document level ($x = 99, n = 143, p < .001$). In contrast to adequacy, fluency ratings in favour of HT are similar in sentence- and document-level evaluation, but subjects find more ties with document-level context as preference for MT drops from 32 to 22 % (Figure 3.3b).

We note that these large effect sizes lead to statistical significance despite modest sample size. Table 3.2 shows detailed results, including those of individual subjects, for all four experimental conditions. Subjects choose between three labels for each item: MT is better than HT (a), HT is better than MT (b), or tie (t). Table 3.3 lists inter-rater agreement. Besides same-label agreement $P(A)$, the proportion of times that two subjects agree on the same label, we calculate Cohen's kappa coefficient

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}, \tag{3.1}$$

where $P(E)$ the likelihood of agreement by chance. We calculate κ, and specifically $P(E)$, as in WMT evaluations (Bojar et al., 2016a, Section 3.3), on the basis of all pairwise ratings across all subjects.

In pairwise rankings of MT outputs, κ coefficients typically centre around .3 (Bojar et al., 2016a). We observe lower inter-rater agreement in three out of four conditions, and attribute this to two reasons. First, the quality of the machine translations produced by Hassan et al. (2018) is high, making it difficult to discriminate from professional translation particularly at the sentence level. Second, we do not provide guidelines detailing error severity (Figure 3.2) and thus assume that subjects have differing interpretations of what constitutes a 'better' or 'worse' translation. Confusion matrices in Table 3.4 indicate that subjects handle ties very differently: in document-level adequacy, for example, S_1 assigns no ties at all, while S_3 rates 15 out of 50 items as ties (Table 3.4b). The assignment of ties is more uniform in documents assessed for fluency (Tables 3.2, 3.4g–l), leading to higher κ in this condition (Table 3.3).

Despite low inter-annotator agreement, the quality control we apply shows that subjects assess items carefully: they only miss 1 out of 40 and 5 out of 128 spam items in the document- and sentence-level conditions overall, respectively, a very low number compared to crowdsourced work (Kittur et al., 2008). All of these misses are ties (i.e., not marking spam items as 'better', but rather equally bad as their counterpart), and 5 out of 9 subjects ($S_{3,5,7-9}$) do not miss a single spam item.

A common procedure in situations where inter-rater agreement is low is to aggregate ratings of different annotators (Graham et al., 2017). As shown in Table 3.1, majority voting leads to clearer discrimination between MT and HT in all conditions, except for sentence-level adequacy.

To allow for external validation and further experimentation, we make all experimental data publicly available.[9]

3.5 Error Analysis

To achieve a finer-grained understanding of what errors the evaluated translations exhibit, we perform a categorisation of 150 randomly sampled sentences based on the error taxonomy used by Hassan et al. (2018).[10] We expand the taxonomy with a Context category, which we use to mark errors that are only apparent in larger context (e.g., regarding poor register choice, or coreference errors), and which do not clearly fit into one of the other categories. Hassan et al. (2018) perform this classification only for the machine-translated outputs, and thus the natural question of whether the mistakes that humans and computers make are qualitatively different is left unanswered.

Our error classification is performed by a bilingual native Chinese and English speaker. Sentences are shown in the context of the document, to make it easier to determine whether the translations were correct based on the context. The analysis is performed on MT and HT, and an alternative human translation which is not discussed further in this chapter.[11] We blind the origin of the translations and randomise the order of experimental items. An example is shown in Figure 3.4.

Results are shown in Table 3.5. We test for significant differences in errors stemming from HT and MT using Fisher's two-tailed exact test, and find significantly larger numbers of errors of the categories of Incorrect Word ($p < .001$) and Named Entity ($p < .05$) in MT, indicating that the MT system is less effective at choosing correct translations for individual words than the human translators. MT also exhibits significantly more Word Order errors ($p < .001$), which is particularly notable given previous reports that NMT systems have led to great increases in reordering accuracy compared to statistical MT systems (Neubig et al., 2015; Bentivogli et al., 2016), demonstrating that the problem of generating correctly ordered output is far from solved for this language pair even in strong NMT systems.

While not statistically significant, likely due to the small number of examples overall, it is noticeable that MT has a higher percentage of Collocation and Context errors, which indicates that the system has more trouble translating words that are dependent on longer-range context. In an article on the female Olympic shooter Zhang Binbin (张彬彬), for example, we see that the MT system was unable to maintain a consistently gendered or correct pronoun (Table 3.6a). Similarly, Figure 3.4 shows an example of lexical coherence in a 6-sentence article about a new app '微信挪车', which HT (column A) consistently trans-

[9] https://github.com/laeubli/parity

[10] Hassan et al.'s (2018) taxonomy is in turn based on, but significantly different than that proposed by Vilar et al. (2006).

[11] See Läubli et al., 2020a. Qinlan Shen, who performed the error categorisation we analyse in this chapter, is a co-author of this paper.

N=50, S_2

S_1		*a*	*t*	*b*
	a	9	4	13
	t	0	0	0
	b	9	0	15

(a) Document, Adequacy

N=50, S_3

S_1		*a*	*t*	*b*
	a	4	9	13
	t	0	0	0
	b	6	6	12

(b) Document, Adequacy

N=50, S_4

S_1		*a*	*t*	*b*
	a	11	1	14
	t	0	0	0
	b	9	2	13

(c) Document, Adequacy

N=50, S_2

S_3		*a*	*t*	*b*
	a	7	1	2
	t	7	1	7
	b	4	2	19

(d) Document, Adequacy

N=50, S_4

S_3		*a*	*t*	*b*
	a	6	1	3
	t	8	0	7
	b	6	2	17

(e) Document, Adequacy

N=50, S_4

S_2		*a*	*t*	*b*
	a	11	2	5
	t	1	1	2
	b	8	0	20

(f) Document, Adequacy

N=50, S_6

S_5		*a*	*t*	*b*
	a	7	2	4
	t	2	4	2
	b	3	8	18

(g) Document, Fluency

N=50, S_7

S_5		*a*	*t*	*b*
	a	6	3	4
	t	2	6	0
	b	3	8	18

(h) Document, Fluency

N=50, S_9

S_5		*a*	*t*	*b*
	a	5	4	4
	t	1	5	2
	b	2	9	18

(i) Document, Fluency

N=50, S_7

S_6		*a*	*t*	*b*
	a	7	3	2
	t	1	7	6
	b	3	7	14

(j) Document, Fluency

N=50, S_6

S_9		*a*	*t*	*b*
	a	5	1	2
	t	4	5	9
	b	3	8	13

(k) Document, Fluency

N=50, S_7

S_9		*a*	*t*	*b*
	a	6	1	1
	t	3	7	8
	b	2	9	13

(l) Document, Fluency

N=104, S_3

S_1		*a*	*t*	*b*
	a	31	6	22
	t	2	0	1
	b	11	10	21

(m) Sentence, Adequacy

N=104, S_8

S_5		*a*	*t*	*b*
	a	16	1	13
	t	10	1	21
	b	10	2	30

(n) Sentence, Fluency

Table 3.4: Confusion matrices: agreement between any two subjects (S_{1-9}) who rated the same items on whether MT is better than HT (*a*), HT is better than MT (*b*), or tie (*t*).

Error Category	Number of Errors		Significance of Difference
	HT	MT	HT vs. MT
Incorrect Word	51	85	•••
Semantics	33	48	
Grammaticality	18	37	••
Missing Word	37	56	•
Semantics	22	34	
Grammaticality	15	22	
Named Entity	16	30	•
Person	1	10	•
Location	5	6	
Organization	4	8	
Event	1	3	
Other	5	7	
Word Order	1	17	•••
Factoid	1	6	
Word Repetition	2	4	
Collocation	15	27	
Unknown Words/Misspellings	0	0	
Context (Register, Coreference, etc.)	6	12	
Any	81	118	•••
Total	129	237	•••

Table 3.5: Blind error classification in MT and HT. Errors represent the number of sentences (out of $N = 150$) that contain at least one error of the respective type. We also report the number of sentences that contain at least one error of any category (Any), and the total number of error categories present in all sentences (Total). Significance levels are denoted by • $p<.05$, •• $p<.01$, and ••• $p<.001$.

lates into 'WeChat Move the Car'; in MT (column B), we find three different translations in the same article: 'Twitter Move Car', 'WeChat mobile', and 'WeChat Move'.

张彬彬和**家人**聚少离多... **父母**说... **张彬彬**很少说**自己的辛苦**，更多的是跟**父母**聊些开心的事。
HT: Zhang Binbin spends little time with **family**... **Her parents** said... Zhang Binbin seldom said **she found things difficult**. More often, **she** would chat about happy things with parents.
MT: Zhang Binbin and **his family** gathered less... Parents said... Zhang Binbin rarely said **their hard work**, more with **their parents** to talk about something happy.

(a) Pronoun

传统习俗引入新亮点"**2016 盂兰文化节**"香港维园开幕敲锣打鼓的音乐、传统的小食、花俏的装饰、人群汹涌的现场。由香港潮属社团总会主办的"**2016 盂兰文化节**"12 日至 14 日在维多利亚公园举办，这是香港最盛大的一场盂兰胜会。
HT: Traditional customs with new highlights - **2016 Ullam Cultural Festival**... The "**2016 Ullam Cultural Festival**" organized by...
MT: Traditional customs introduce new bright spot "**2016 Ullambana Cultural Festival**" ... Organised by the Federation of Teochew Societies in Hong Kong, the "**2016 Python Cultural Festival**" is ...

(b) Named entity

Table 3.6: Examples of inconsistent translation across sentences in MT.

3.6 Discussion

3.6.1 How should strong MT systems be evaluated?

Our findings call for revisiting several design choices made in human translation quality assessments.

Task

Hassan et al. argue that the outputs of their 'research systems are indistinguishable from human translations' (Hassan et al., 2018, p. 15), but in source-based DA (Section 2.2.3), subjects are never asked to directly compare HT and MT: the task is to assign an absolute score to one English translation of one Chinese source sentence at a time, and this task is repeated several times with different source sentences and translations from either HT or

市民在日常出行中，发现爱车被陌生车辆阻碍了，在联系不上陌生车辆司机的情况下，可以使用“微信挪车”功能解决这一困扰。8月11日起，西安交警微信服务号“西安交警”推出“微信挪车”服务。这项服务推出后，日常生活中，市民如遇陌生车辆在驾驶人不在现场的情况下阻碍自己车辆行驶时，就可通过使用“微信挪车”功能解决此类问题。这个新服务功能一是快捷，节省行政成本；二是尊重、保护公民个人隐私；三是全天候、全时段服务。只要市民有微信，就可以非常便捷地自行操作，这种方式相对能使移车双方都能比较友好轻松地“相遇”，缓解停车难问题，并可减少涉车的治安案件发生。另一方面，对于拒不移车、违法停放或妨害通行的车主，西安交管部门表示，交警届时将会根据实际情况采取不同的处置措施，视情况给予处理。

A

A citizen whose car is obstructed by vehicle and is unable to contact the owner of the obstructing vehicle can use the "WeChat Move the Car" function to address the issue. The Xi'an Traffic Police WeChat official account "Xi'an Jiaojing" released the "WeChat Move the Car" service since August 11. Once the service was released, a fellow citizen whose car was obstructed by another vehicle and where the driver of the vehicle was not present, the citizen could use the "WeChat Move the Car" function to address the issue. The new service firstly provides convenience and reduces administrative costs; secondly, it respects and protects public privacy of citizens; and finally, it offers services at any time of the day. This can be operated independently and very conveniently by any citizen with WeChat. This method allows both parties in the vehicle moving issue to "meet" in a more amiable and light-hearted manner, alleviate issues of parking difficulties, and reduce public security incidents relating to vehicles. On the other hand, for vehicle owners who refuse to move their vehicle, commit parking violations, or obstruct the passage of others, the Xi'an traffic control department

B

Members of the public who find their cars obstructed by unfamiliar vehicles during their daily journeys can use the "Twitter Move Car" feature to address this distress when the driver of the unfamiliar vehicle cannot be reached. On August 11, Xi'an traffic police WeChat service number "Xi'an traffic police" launched "WeChat mobile" service. With the launch of the service, members of the public can tackle such problems in their daily lives by using the "WeChat Move" feature when an unfamiliar vehicle obstructs the movement of their vehicle while the driver is not at the scene. The first function of this new service is to be fast and save administrative cost; the second is to respect and protect the individual privacy of citizens; and the third is all-weather, full time service. As long as citizens have WeChat, they can be very convenient to operate on their own, which can relatively make both sides of the car can be relatively friendly and easy to "meet" to ease parking problems, and can reduce the occurrence of vehicle-related security cases. On the other hand, for those who refuse to drive, illegally park or obstruct the passage of vehicles, Xi'an traffic control department

C

In the daily drive, if a citizen is blocked by a randomly parked car, and cannot find the owner, they can use the "WeChat Move Car" function to solve this issue. "The Xi'an Traffic Police" launched the "WeChat Move Car" service on the WeChat app from August 11. After the launch of this service, in the event that unknown vehicles without the driver is parked and in the way, citizens can use their “WeChat Move Car” function to solve such problems. This new service function is fast and saves administrative costs. Secondly, it respects and protects the privacy of citizens. Thirdly, the service can be provided anytime regardless of weather. As long as the citizens have WeChat, they can do this on their own and very easy and convenient. This way, both parties can "meet each" easier and resolve problems relating to parking, and reduce public security incidents involving vehicles. On the other hand, Xi'an traffic management department said that the traffic police will take different measures based on the actual situation, and handle them on a case-by-case basis.

Figure 3.4: An experimental item as shown to the human annotator for blind error categorisation. Translations A and B are cropped.

	WMT 2018 (Bojar et al., 2018)	Hassan et al., 2018	This Chapter (Läubli et al., 2018b)	Toral et al., 2018	WMT 2019: SR+DC[13] (Barrault et al., 2019)	Toral, 2020	Fischer and Läubli, 2020
Reassessment of:			Hassan et al., 2018	Hassan et al., 2018		WMT 2019	
Rated item:	Sentence	Sentence	Document	Sentence	Sentence	Sentence	Sentence
On-screen context:	None	None	Full Document	Previous and next sentence	None	Previous and next sentence	Full Document
Order of sentences:	Random	Random	Document order	Document order	Document order	Document order	Document order
Task:	Direct Assessment	Direct Assessment	Relative Ranking	Relative Ranking	Direct Assessment	Relative Ranking	Error Categorisation
Translations per item:	1	1	2	3	1	2–3	1
Finding:	Parity (EN→CZ)	Parity (ZH→EN)	No parity (ZH→EN)	No parity (ZH→EN)	Parity (DE→EN, EN→RU); super-human performance (EN→DE)	No parity (DE→EN, EN→RU); parity (EN→DE)	Similar number of errors (DE→FR, DE→IT, DE→EN)

Table 3.7: Summary of human–machine parity assessments.

MT. In this way, HT and MT are rated independently, and parity is assumed if the scores for the former do not significantly differ from the scores of the latter. We use relative ranking instead of DA, always showing a pair of translations: one produced by HT, one by MT (in random order). While DA has some advantages over relative ranking – notably: quantifying the degree to which a translation is preferred over another (Graham et al., 2017) – the outcomes of DA and relative ranking correlate strongly (Bojar et al., 2016a), and we consider a direct comparison of HT and MT (relative ranking) more meaningful than a comparison of independently obtained scores (DA) for the same source texts if the aim is to assess human–machine parity. While our design choice is in line with other reassessments of parity claims (Toral et al., 2018; Toral, 2020), we are not aware of a study that empirically tests the impact of using relative ranking over DA in a human–machine parity assessment.

Materials

In our reassessment of Hassan et al.'s (2018) evaluation, we only used English translations of source texts that were originally written in Chinese (Section 3.3.2). While it has been common practice in MT evaluation to use the same test set in both translation directions (e.g., Bojar et al., 2017, 2018), we consider a direct comparison between human 'translation' and MT hard to interpret if one is in fact the original English text, and the other an automatic translation into English of a human translation into Chinese. According to Laviosa (1998), translated texts differ from their originals in that they are simpler, more explicit, and more normalised. For example, the synonyms used in an original text may be replaced by a single translation. These differences are referred to as translationese, and have been shown to affect translation quality in the field of MT (Kurokawa et al., 2009; Daems et al., 2017). Toral et al. (2018) show that this also holds with Hassan et al.'s (2018) translations: subjects prefer HT over MT with source texts originally written in Chinese (i.e., the texts we consider in our experiment), but not with source texts originally written in English.

Naturally, even original translations can differ in quality. Läubli et al. (2020a) examine professional translations of the Chinese articles in the WMT 2017 test set that were optimised for target language fluency. According to a blind error analysis, these translations contain fewer grammatical errors, but significantly more omissions; and in a relative ranking experiment with the experimental design proposed in this chapter (Section 3.3), professional translators rate them significantly better in terms of fluency, but not significantly different from Hassan et al.'s (2018) MT in terms of adequacy, in both the Sentence and Document conditions. Along with other examinations (Freitag et al., 2020), this finding shows that the nature of human reference translations used or specifically created for MT evaluations can impact their outcome, and highlights the importance of selecting or having experimental materials created with care. Toral (2020) raises the question of whether strong MT systems should be compared to 'average' or 'champion' translators, and brings up an interesting analogy: in other areas, such as playing board games like chess, it would

be odd to claim that a computer program achieves human parity if it beats an average chess player, but rather credible if it beats a world champion. Since world champions are far outnumbered by average chess players, on the other hand, achieving parity with the latter may still be an impactful achievement.

Subjects

To optimise cost, MT quality is typically assessed by means of crowdsourcing. Combined ratings of bilingual crowd workers have been shown to be more reliable than automatic metrics (Section 2.2.3) and 'very similar' to ratings produced by 'experts'[12] (Callison-Burch, 2009). Graham et al. (2017) compare crowdsourced to 'expert' ratings of statistical machine translations from WMT 2012, concluding that, with proper quality control, 'machine translation systems can indeed be evaluated by the crowd alone.' We chose to involve professional translators rather than crowd workers because we assumed that these findings would not carry over to translations produced by strong NMT systems where, due to increased fluency, errors are more difficult to identify (Castilho et al., 2017a). Toral et al. (2018) confirm this assumption empirically: while professional Chinese to English translators prefer HT over MT, non-experts (NLP researchers native in Chinese with an advanced level of English) do not.

Nevertheless, hiring professional translators to evaluate MT output is expensive, and it could be argued that since most consumers of MT will not be professional translators, it may make sense to involve these consumers in an evaluation. Simply put: professionals may be better at distinguishing nuances in translations, but these nuances may not matter to end users. On the other hand, it can be assumed that consumers will be glad to choose one MT system over another if professional translators find it to be better in aspects that consumers cannot assess themselves. While the choice of experimental subjects will depend on various factors such as use cases and budgetary constraints in practical scenarios, we believe that extraordinary claims such as MT achieving parity with professional HT should be based on judgement by experts. Referring back to to Toral et al.'s analogy, it would be odd to assess whether a program can beat a skilled human at chess by asking individuals who do not know how chess works.

Procedure

In line with WMT 2018, Hassan et al. (2018) present the sentences to be evaluated in random order, and do not show any surrounding sentences. This prevents subjects from identifying errors related to document-level coherence and cohesion, such as wrong pronouns (Table 3.6a), and gives an unfair disadvantage to HT, which contains less of these errors (Table 3.5). We present full documents in the Document conditions of our experiment instead, and show that this results in significantly lower preference for MT (Sec-

[12] We note that 'experts' here are computational linguists who develop MT systems and may not be expert translators.

tion 3.4). As we collect judgements for documents rather than sentences, one disadvantage of our approach is that we can obtain far fewer judgements in an experiment with a given budget, resulting in low statistical power compared to sentence-level experiments (Graham et al., 2019). In steering a middle course between sample size and validity in terms of linguistic context, other experimental designs use segment-by-segment presentation in document order (Barrault et al., 2019, SR+DC), display the previous and next sentence for each sentence to be rated (Toral et al., 2018), provide full documents as separate text files (Toral, 2020), or elicit segment-level ratings while showing full documents (Fischer and Läubli, 2020). In line with our findings, a comparative analysis of these studies suggests that showing more linguistic context leads to lower preference for MT over HT (Table 3.7).

3.6.2 Has MT reached parity with professional HT?

Our findings and concurrent work by Toral et al. (2018) show that Hassan et al.'s (2018) strong Chinese to English MT is indistinguishable from professional HT to crowd workers who score isolated sentences in random order, but not to professional translators who directly compare MT and HT of full documents (this chapter) or sentences shown in document order (Toral et al., 2018). While Hassan et al. (2018) follow best practices in MT evaluation, our findings call for revisiting these practices: as MT quality improves, translations are becoming harder to discriminate in terms of quality, and it may be time to shift towards document-level evaluation, which gives subjects more context to understand the original text and its translation, and also exposes translation errors related to discourse phenomena which remain invisible in an evaluation of isolated sentences. This call has been heeded at WMT 2019 (Barrault et al., 2019). The large-scale human evaluation of the News Translation Task still uses DA and crowd workers, but does no longer present experimental items in random order: in the DR+DC condition, subjects see and score full documents; in the SR+SC condition, subjects see and score single sentences one-by-one in document order. Furthermore, the test sets only contain translations of original news articles, i.e., no translationese.

While this arguably marks an improvement in MT evaluation, MT quality has improved, too. In the SR+DC condition[13] at WMT 2019, the scores of 'many systems are tied with human performance' in German to English, English to German, and English to Russian, and Ng et al.'s (2019) system 'achieves super-human translation performance' in English to German (Barrault et al., 2019, p. 23). Toral (2020) shows that with professional translators as subjects and relative ranking instead of DA with more on-screen context, these findings are not substantiated – however, HT is not preferred significantly over Ng et al.'s (2019) MT in English to German, marking a finding of 'parity' assessed with an evaluation design adhering to many of the recommendations put forward in this chapter (Section 3.7). The strong performance of current MT systems is also highlighted in Fisc-

[13] Sentence rating (SR) with document context (DC), i.e., 'the assessment of individual segments which are nevertheless provided in their natural order as they appear in the document' (Barrault et al., 2019).

her and Läubli's (2020) document-level evaluation of domain-specific MT and HT from German to French, Italian, and English, where professional translators specialised in the insurance domain find similar numbers of omissions and terminology errors in both MT and HT.

However, the absence of a 'statistically significant difference between human quality scores for a test set of candidate translations from an MT system and the scores for the corresponding human translations' – Hassan et al.'s (2018) definition of 'parity' – is no evidence for *equivalence* between human and machine translation. Terms like 'parity' and 'super-human translation performance' (Barrault et al., 2019) are problematic because they imply such equivalence, but naturally, machines achieve super-human performance in some aspects of translation, just like humans achieve super-machine performance in others. In terms of speed, for example, machines have long outperformed human professionals; machines, on the other hand, cannot take personal responsibility for the correctness of a translation, which is required in many applications. Ultimately, we believe that further assessments will need to focus on what specifically professional translators can do better than MT systems – and vice versa – rather than grading their 'output quality' as such.

3.7 Summary and Recommendations

Our reassessment of Hassan et al.'s (2018) investigation of Chinese to English news translation shows that design choices in translation quality evaluation experiments can have a strong impact on their outcome. In a source-based DA task with bilingual crowd workers and the WMT 2017 test set, in which some documents are translationese, scores for MT and HT are not significantly different, through which Hassan et al. (2018) argue to have 'achieved human parity in translating text in the news domain' with a strong MT system. We use the same human and machine translations (except for the translationese documents) in a relative ranking experiment with professional translators, and find a significant preference for HT over MT when full documents are evaluated instead of isolated sentences.

Läubli et al. (2020a) synthesise the findings presented in this chapter with concurrent work by Toral et al. (2018), and recommend a set of evaluation design choices that we believe are needed for assessing human–machine parity, and will strengthen the human evaluation of MT in general. The recommendation that follows from this chapter in particular is to evaluate documents instead of sentences. When evaluating sentences in random order, professional translators judge MT more favourably (Section 3.4) as they cannot identify errors related to textual coherence and cohesion, such as different translations of the same product name (Section 3.5). Our experiment shows that using full documents as experimental items increases the rating gap between HT and MT, and refutes Hassan et al.'s (2018) claim of human–machine parity in Chinese to English translation. Our findings have influenced the human evaluation campaign at WMT (Barrault et al.,

2019, p. 16), which is now based on sentences presented in document order (SR+DC) and, for some languages, full documents (DR+DC).

Acknowledgements

We are indebted to Xin Sennrich and Qinlan Shen for their help with the analysis of the translation errors discussed in Section 3.5.

Chapter 4

Translator Requirements for Text Presentation in CAT Tools

If we accept our conclusion drawn in Chapter 3, translations produced by strong MT systems have not reached – but may be approaching – the quality of translations produced by professional translators. However, even if professional translators decide not to use MT (e.g., for post-editing) in their daily work, other forms of translation technology have long become indispensable. Translation aids such as TMs and TBs, typically embedded in a bilingual text editor referred to as CAT tool (Section 2.3.2), provide translation suggestions that influence translation decisions made by professionals. If design choices made in CAT tools hinder translators from producing optimal translations, this will disadvantage them not only in human-vs.-machine assessments, but in their daily work altogether.

Translation scholars and practitioners have voiced concerns about the usability of CAT tools (Section 4.1), and in this chapter, we present a qualitative survey among professionals with a focus on visual context. Since the way in which translations are shown to translators in a quality evaluation experiment has an impact on how good these translations are judged to be, we assume that the same may hold in CAT tools: if translators can focus on document-level context, they may be faster at producing translations and/or able to achieve better quality. We ask participants how they use and what advantages and disadvantages they see in translation technology that is currently available (Section 4.3.1), elicit ideations of what a perfect CAT tool could look like (Section 4.3.2), and gather feedback on document-level editing (Section 4.3.3), from which we distil design recommendations in Section 4.4.

4.1 Background

Most professional translators work with CAT tools (Zaretskaya, 2015; Schneider et al., 2018) but treat them with 'suspicion or disinterest' (O'Brien et al., 2017). Among numerous explanations offered in published research, two reasons stand out in particular:

dehumanisation and poor usability.

O'Brien (2012) states that the increasing need or obligation to use technology impacts the status of the translation profession. In having to resolve problems introduced (in the case of MT) or propagated (in the case of TM) by translation technology, the role of the professional translator becomes that of a fixer,[1] which 'irks some translators to such a degree that they refuse to interact with the technology' entirely (O'Brien, 2012). Translators are not willing, and often not trained (Belam and Lewis, 2002), to fix errors that may differ fundamentally from errors that humans would generate, particularly if they are remunerated at lower rates compared to translation from scratch. O'Brien further mentions fear of being replaced by a machine, a factor also identified in Cadwell et al.'s (2018) focus group meetings with professional translators: of the 90 participants, more than 50 % said that a reason for not using MT was 'fear (e.g. of the unknown, of being replaced by a machine)', and more than 25 % said that 'using MT devalues the translator's work'.

Another reason for resistance towards CAT tools is poor usability. O'Brien et al. (2017) analyse response data from Ehrensberger-Dow et al.'s (2016) survey taken by more than 1,800 translators, and identify several CAT tool features that cause irritation.[2] CAT tools are complex from a technical perspective because they combine multiple backend systems systems to provide translation suggestions based on technical constraints and user-defined rules (Section 2.3.2), and O'Brien et al.'s (2017) analysis reveals that developers have not succeeded in reducing this complexity for end users: 'The most common issue mentioned was the Complexity of the UI, which, on further analysis, indicated a lack of an intuitive navigation system, lack of user friendliness and a need for too many mouse clicks to perform actions within the tool' (ibid.). The second most common issue mentioned was segmentation, i.e., 'Issues with segmentation feature such as problems merging segments, autopropagation and irritation caused by segmented view of text' (ibid.). The last problem in particular may relate to our finding that the presence or absence of linguistic context impacts perceived translation quality (Chapter 3). Most CAT tools in wide use separate documents into sentences and arrange them in a side-by-side, spreadsheet-like view (Figure 2.12), which limits the amount of linguistic context visible to translators when translating a document. This segment-by-segment presentation of texts has been described as 'unnatural' (Dragsted, 2006) and 'a barrier to creativity' (LeBlanc, 2013), and found to result in an 'obstructed view of the text, which in turn disrupts the [translation] workflow' (O'Brien et al., 2017).

Nevertheless, the issue of text segmentation has not been addressed in CAT tools. On the contrary, 'the [CAT] tools used for the last decades to recycle human translation are being adopted also for the task of post-editing MT output' (Moorkens and O'Brien, 2017) – CAT tools are extended to accommodate more and more functionality, but the user in-

[1] Alluding to Krings's (1994) habilitation thesis entitled 'Texte reparieren' (*Repairing Texts*), a seminal investigation into MT post-editing.

[2] O'Brien et al. (2017) base their definition of irritation on the concept of cognitive friction, 'the resistance encountered by a human intellect when it engages with a complex system of rules that change as the problem changes' (Cooper, 2004).

terface remains unchanged. In a survey among more than 200 professional translators, Moorkens and O'Brien (2017) assess user interface needs for MT post-editing, and find that 38 % of their participants use Microsoft Word rather than a CAT tool for post-editing. The implication is that CAT tools exhibit a degree of complexity that causes many translators to give up translation suggestions and functionality (which Word does not offer) in favour of a tool they find easy to use.

In the survey described in the remainder of this chapter, we seek insights into what causes professional translators to use or abandon CAT tools, and if a display of full documents – as seen in regular word processors – could resolve some of the problems currently found with segment-by-segment presentation. We recruit fewer participants than most of the aforementioned studies, which allows us to elicit qualitative feedback and engage in open-ended discussions in one-to-one sessions.

4.2 Survey Methods

We conduct semi-structured interviews on translation technology with eight professional translators. In the first part, participants are asked what they like and dislike about CAT tools currently available. In the second part, we ask what their CAT tool of choice would look like if there were no technical and budgetary restrictions. Lastly, we introduce the concept of document-level editing: a CAT tool that lets translators manipulate documents as a whole rather than individual segments. Our aim is to assess if this concept is viable, and to elicit feedback to inform the design of prototypes.

4.2.1 Design

We adopt a concept testing methodology (Moore, 1982) to elicit qualitative feedback on using document-level editors for translation. In semi-structured interviews, participants are walked through a series of closed and open-ended questions (Section 4.2.4). We reassure participants that any response will be anonymized, and that participation in the study will in no way influence their professional relationship with their employer. We also emphasise that participants are not being tested and, at the beginning as well as throughout the interview, encourage them to be critical.

4.2.2 Materials

We conduct remote interviews using a web-based communications platform (audio only). All participants consent for us to have the conversation with them recorded. In the last part of the interview, we make use of an online whiteboard for collaborative sketching. Participants can use either their personal computer, tablet, or phone for both the audio call and using the whiteboard.

4.2.3 Participants

We recruit 8 participants from a multinational translation agency's pool of professional freelancers. They are recommended by a project manager (convenience sampling) and invited to participate via an internal communications platform. We do not offer compensation.

Participants have been working as full-time professional translators between 0.5 and 13 years (avg=4.4) and are well-acquainted with computer-aided translation technology (use between 0.5 and 9 years; avg=3.3), notably because the aforementioned translation agency requires them to work with a web-based CAT tool that integrates adaptive MT. As such, the profile of our participants will differ from that of related studies where several individuals have no experience in post-editing MT (e.g., 25 % in Moorkens and O'Brien, 2017).

4.2.4 Procedure

After a pilot run on 31 July 2018, all interviews are conducted between 1–3 August 2018. The duration varies between 30 and 50 minutes.

4.3 Results

4.3.1 Current State

All participants report to use TMs, TBs, MT from within their CAT tool (MT: Integrated), and tools for quality assurance (QA) either sometimes or regularly (Table 4.1). This is above-average compared to large-scale surveys on translation technology adoption among professionals (Section 4.1) and likely owed to the fact that all participants regularly work with a web-based CAT tool, provided by the translation agency they work for, that includes all of the features listed above. As such, our sample is not representative of the overall population of professional translators.

To gain more insights into how participants currently use computers for translation, as well as what they like and dislike about it, we ask them to think back to the translations they have produced within the last 12 months. All translator feedback described in the remainder of this section refers to this period of time. We ask participants for feedback on any CAT tool they have been working with. However, they often refer to the particular CAT tool provided by their agency, likely because of regular use and the fact that someone from this agency referred them for the interview.

	Regular	Sometimes	Never
Translation Memory (TM)	7	1	0
Terminology Database (TB)	4	4	0
Machine Translation (MT): Standalone	1	3	4
Machine Translation (MT): Integrated	6	2	0
Term Extraction	0	1	7
Quality Assurance (QA)	8	0	0

Table 4.1: Use of translation technology among participants.

Translation Without Electronic Devices

We ask participants whether they have, within the last 12 months, produced translations without the use of any electronic device. We want to hear if translators would decide to refrain from such devices in certain situations, and whether shortcomings of current technology motivate this decision.

However, none of the participants produced a single translation without the use of electronic devices in the last year.

Translation Without CAT Tools

We then asked participants whether they have produced some of their translations with a computer, but no computer-aided translation software. Moorkens and O'Brien (2017) note that 91 out of 246 professional translators they surveyed (38 %) use Microsoft Word for post-editing, and our hope is to learn why translators would stay away from software specifically designed to assist with this process.

As we ask for a rough estimate of translation jobs completed without CAT tools in the last 12 months, answers vary between 0 % (2) or 1 % (1) to 10–20 % (4) and up to 75 % in one case (avg=17.0 %). Participants report to use Word, Excel, Google Docs, Google Sheets, and E-Mail clients when not using CAT tools.

Participants mention several reasons for not using CAT tools. The most prominent is administrative overhead: for short translations and/or when fast turnaround is vital, uploading documents to, creating projects in, and exporting translations from CAT tools would take more time than it saves. Participants also mention that some file formats such as PDF do not work well with CAT tools, or that they will not use a CAT tool if none is provided by their client. P5 mentions that reviewing translations is easier outside of CAT tools. To illustrate that CAT tools are more suitable for some text genres than others, P1 states that 'if I'm doing a very specific medical text, the CAT tool doesn't really get there anyway because there are so many technical terms so that it looks like a big mess anyway…

so that I really don't think I'd have anything to gain with a CAT tool.'

When asked 'Do you like translating without CAT tools?', participants mostly answer in the affirmative. Recurrent themes brought up are non-distraction, freedom, and ownership. P3 states that 'there's a sense of no interruptions, there's not a cluttered screen in word editors', and P5 mentions that 'there's more focus' and 'you have more ownership over the creative process' when not using a CAT tool. Referring to the presentation of texts in regular word processors, P3 says that 'it doesn't interfere – it's a white sheet.' P5 makes a similar comment mentioning 'the empty page'.

Translation Memories

Participants use TMs for between 10 and 100 % of translation jobs completed within the last year (avg=49.3 %; N=7, one translator 'can't tell'). We ask participants what they like and dislike about working with TMs.

Participants highlight productivity and familiarisation with new topics. P2 likes 'that you don't actually have to reinvent the wheel every single time'. P3 finds TMs useful to immerse into a topic one is not familiar with, and P6 tells they would often do a concordance search within the TM rather than using an online search engine.

On the downside, participants note that TMs may slow them down occasionally. Consistent with feedback about what participants like about working without a CAT tool – most of which include some sort of TM – some feel to have less freedom and a reduced sense of ownership when working with TMs. P5 sometimes feels like 'losing my power or my ownership. … You wanna say it this way, but it says "no, say it that way"'. Similarly, P6 talks about priming effects:

> The flipside of having ideas [i.e., TM matches] is that it can be somewhat more difficult to come up with your own original translation for something. So if you don't really like a particular translation that has been done in the past, it might be difficult to step away from that translation that was part of the translation memory and come up with your own. So it has a potential to sort of stifle originality, creativity.

P3 highlights that TMs are more suitable in some cases than others. For example, they were not suitable for transcreation. P4 describes TMs as 'pretty nifty, though a pain in the ass to actually assemble.'

Termbases

Participants have used TBs less frequently than TMs in jobs completed within the last 12 months (between 5 and 50 %, avg=30.0 %). Many note that they would use them whenever provided by the client.

P1 and P5 mention that TBs are helpful to stick to corporate terminology. Other than that, negative comments prevail. P5 finds TBs to seem 'dumb' as they suggest very obvious translations. P6 is more concerned about false positives: when a TB suggestion is ignored because it does not fit in a particular context, this may lead to problems in QA. This participant also felt that there is a tendency to integrate online resources (including TBs) into CAT tools, causing problems with connectivity (i.e., latency, unavailability).

While many comments centre around content, we also receive some feedback about interaction. Notably, P7 finds it 'annoying' when terms have to be looked up actively (i.e., the translator triggers the search) because it interrupts their workflow.

Standalone Machine Translation

Half of the participants report using standalone MT – such as Google Translate – sometimes or regularly (Table 4.1). Usage varies between 0 and 30 % (avg=6.4 %), again with respect to translation jobs completed within the last 12 months.

P4 uses Google Translate among other tools (such as Linguee) to see translation variants for a certain term or phrase. P5 says 'I do a lot of MTPE [...]. Most times you have that machine translation that the client gave you that they run through their own [engines] and it's just so bad.' Replacing MT provided by clients with MT produced by Google Translate would sometimes make them more productive.

Other participants say that they do not use standalone MT because of low quality or the availability of integrated MT (see below).

Integrated Machine Translation

Being required to use their agency's CAT tool with integrated MT (see above), all participants make use of this feature sometimes or regularly. In the last 12 months, participants have used it in between 40 and 90 % of their jobs (avg=53.8 %).

Participants like that integrated MT makes them more productive, provides translation hints, and – in case of adaptive MT – learns from their edits. P2 points out that 'it's definitely faster' than translating from scratch, and others agree. P3 finds that integrated MT 'functions like a synonym provider' in that 'it works like an alternative mind [...] sending you suggestions.' With regards to adaptive MT in particular, P1 likes that suggestions are adapted to translators over time.

However, P1 criticises that adaptation can be slow: 'Sometimes you're surprised that some things it can't learn more quickly while other expressions it learns immediately.' Participants also raise the issue of low quality. P5 compares their agency's adaptive MT to Google Translate, joking that the former 'is like Google who's been drinking for like three nights, you know... binge drinking.' On a more serious note, participants bemoan repetitions of words (P1), suggestion of non-words (P2, P5, P7), and use of inappropriate

language (P5). As a result, 'you have to be aware of and pay attention to what the machine suggests you' (P2). P6 is unhappy with the fact that MT suggestions are displayed all the time, regardless of how good and suitable they are (lack of quality estimation). Apart from that, P2 and P3 complain about latency: 'If I see my fingers go faster, I use my fingers; if I see the TM is faster, I use it' (P3).

4.3.2 Ideation

In the second part of the interview, we ask participants what a perfect CAT tool would look like. We phrase the question as follows: 'If suddenly a team of translation technology developers walked into your house and asked about the CAT tool of your dreams, what would you tell them to build?' We encourage participants to think bold and leave budgetary or technical limitations aside. Participants have access to an online whiteboard for collaborative sketching with the interviewer, and are told that they are free to use it or not.

Although the primary motivation of our survey is to elicit feedback on document-level editing, we do not share or even mention any such concept before or at this point as we want to see if participants will initiate the topic themselves.

The features that participants wish for are listed in Table 4.2; wording and aggregation are ours. We identify three main themes: knowledge acquisition, translation alternatives, and visual context.

Knowledge Acquisition

Participants ask for integrated tools for knowledge acquisition. P4, for example, tells that they are currently translating materials for the printing industry which they are not familiar with. When confronted with terms such as 'flat lay', they have to find out what that was before translating it. To this end, participants often use Google Images or Wikipedia. They wish to be able to use these resources from within their CAT tool as 'it's time-consuming to hop around different things to research one term for a given context' (P4). More specifically, participants wish for an integrated dictionary, an integrated image search, and a small integrated web browser (Figure 4.1).

Translation Alternatives

Participants also tell us that they often spend time looking for the best way to translate a certain term or phrase in a specific context. Some use multiple external resources, including Linguee and Google Translate, and would like to see these integrated because 'then you don't have to open multiple tabs and you don't have to shift between them and [...] technically you could suffice with a single screen' (P2). Others would like to get multiple suggestions (n-best translations) with an easy-to-use mechanism to trigger and

Functionality:

- Integration of external resources
  - Dictionary (P2)
  - Small web browser for research (P3, P4)
  - Image search (P4)
  - Link image (source material) to segment (P7)
- View without segmentation
  - of source material (P2, P3, P8)
  - of end result (P2, P3, P8)
- Source text
  - Edit source text (P8)
- Target text (user input)
  - Voice recognition (P6)
  - Rich text editing tools (P8)
  - Hotkey for selecting n^{th} MT word (P2, P3)
  - Set character limitation (P7)
- MT
  - Show alternatives (P1, P2)
- Misc
  - Progress indicator (P2)

Quality:

- Look
  - Clean (P3)
  - Uncluttered (P3)
- Speed
  - Low latency (P5)

Table 4.2: Features mentioned by participants when asked what a perfect CAT tool would entail.

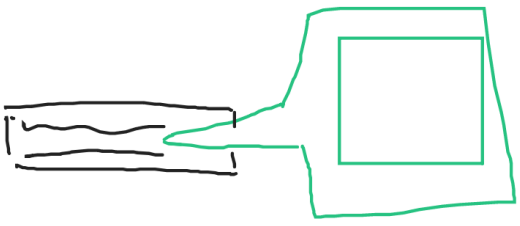


Figure 4.1: Integration of a small web browser for research (P3).

Figure 4.2: Interviewer's visualisation of segment- (left) and document-level (right) CAT tools.

select them (e.g., hotkeys). P7 wants word or phrase level alternatives 'to be displayed all at the same time [...] in like a mini mini [sic] dropdown menu and then with a nifty hotkey you select [...] the one you want.'

Visual Context

P8 states that an ideal editor 'should look like a Word document but with some addiction on [sic] a CAT tool.' A number of participants tell us that it is hard for them to translate without knowing what the source and target document looks like. P3 says that 'we spend a lot of time thinking like "What, where does this come from?"', and that 'the ideal thing would be to have a visual context for the text, like you know where that segment is in the final page.' Similarly, P2 would have included some sort of preview in the CAT tool of their dreams. P8 almost apologised for asking to 'see the whole text' during translation: 'I know it sounds small, but it would be really useful.'

4.3.3 Concept Testing

In the last part of the survey, we focus on translation with a theorised document-level CAT tool. If participants do not bring up the topic themselves, we steer the conversation towards it by saying we were 'thinking about translation software that lets you focus more on documents as a whole rather than individual segments.'

We then illustrate the concept as we thought of it using the online whiteboard (Figure 4.2). We start by sketching out how documents are split into sentences with current translation editors, how each sentence is put into a separate box, and how only few boxes fit into the user's screen and are thus visible at the same time. We then contrast this with a screen showing two full pages: the left one containing the source text, the right one being empty. In that sense, our drawing resembles the 'print layout' available in Microsoft Word or Google Sheets, but with two parallel documents.

The rest of the interview is open-ended. We ask participants what they like and dislike about the concept, what opportunities and limitations that they see, and follow up on individual remarks.

Praise and Opportunities

First reactions are positive from seven out of eight participants, ranging from 'very nice' (P2) to 'amazing' (P5). When asked what they like about the concept, participants highlight the potential for better translation quality and ease of use. Some participants think that translations produced with such an editor 'would be more true to the source' (P4), and that a translated document would 'seem less as a translation than when you do the segment for segment thing instead' (P6). P8 says it would help solve some 'really hard trouble' with the CAT tools they currently use: 'Sometimes [...] when I'm going to send my files back to my clients they struggle to [put the translations into] the same layout.' Along the same lines, P7 likes that 'you could see whether you're translating a title, a subtitle [...]'. P6 called the concept 'quite a big leap' and pointed out that it might ease the merging and splitting of segments:

> As a translator in [the CAT tool Trados] Studio I'm just looking at that one segment. Of course I know the segments that are around it, but I think, uhm, I would imagine that if you have the entire document as a thing that you're translating you could maybe move some things from one sentence to another whereas in [Trados] Studio you could never do that. Or you can, but it's a lot of annoying stuff to merge segments together or split them. You're basically tied to whatever the system has thought is a good way to segmentise the text. And that's not how a person will read that text eventually, it's sort of a mismatch between how the text will be used and how the translation is done.

As for opportunities, P5 thinks the concept is 'like you're using [the CAT tool] MemoQ, but at the same time you get to see the final product which, you know, when you use either MemoQ or Trados you're going blind until you're done and you export it and you're praying that it's gonna be OK.' P3 notes that document-level editing could be helpful for reviewing in particular.

Criticism and Limitations

One participant's first reaction was negative, saying that 'instinctively it feels like a step back for me' (P1). The concept reminds them of what translation was like before using CAT tools, when jumping back and forth with their eyes between two documents felt tiring; they find it helpful that CAT tools break texts down into 'this nice long list'. In the same vein, P5 says that 'you need segmentation sometimes to just better focus', and P3 found segment-by-segment presentation useful to 'order yourself' and work bit by bit. In that sense, lack of orientation is the main reservation among participants. P6 mentiones that even working with long sentences can be hard:

> The one thing that I can compare this [document-level editing] with is whenever I have a large segment in [Trados] Studio, and when that happens I'm never happy because it's annoying, you tend to use lose track of where you are because at some point there's gonna be a mismatch between, uhm, oh in the English segment I'm like in the third row whereas in Dutch it's a longer language so I'm probably gonna be on the fifth row already. It's gonna be like not matching up correctly so you sort of check back all the time like 'where was I?'"

Participants also note that document-level CAT tools would be unsuitable for certain text types, such as a list of keywords or colours. P4 says a document-level editor would only be helpful 'if you're dealing with a whole document like an article or a medical record or something as a CV, something that is actually gonna have sentences on it.'

4.4 Discussion and Design Implications

We discuss and distil design implications from the qualitative response data presented in the previous section. Our aim is to inform the design of (prototypes for) document-level CAT tools.

4.4.1 Help Users Orientate

Several participants point out that it might be difficult to keep track of where they are when switching back and forth between source and target text (Section 4.3.3). Whenever interviewees expressed this concern, we asked them if highlighting the source segment that is currently being translated would mitigate this problem. While some agree, P5 thinks that this would be more distracting than useful. P4 suggests highlighting sections or paragraphs rather than sentences, an interesting and possibly less error-prone option with automatic sentence alignment.

Another participant (P8) suggests placing source and target texts on top and bottom rather than left and right, as illustrated to the left in Figure 4.3. This would be consistent with the design decision in Green et al.'s (2014a) segment-level CAT tool, where reducing gaze shift motivated a top-and-bottom presentation of source and target segments.

4.4.2 Allow Focus on Individual Segments

P5 states that 'you need segmentation sometimes to just better focus.' Enabling translators to switch between segment and document level views would probably be most effective (P2, P5, P7), also because the two are suitable for different text types (P4).

A 'focus mode' could be another option. It could allow translators to black everything out, except for the segment or paragraph they are working on.

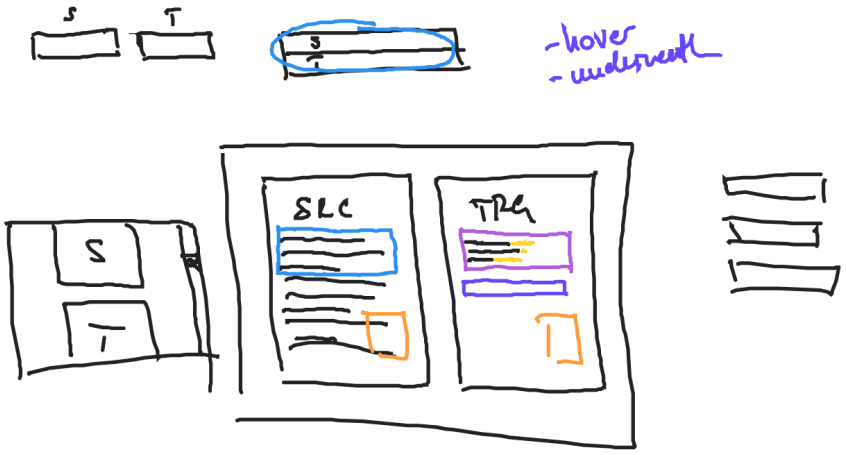


Figure 4.3: Collaborative sketching with P8.

4.4.3 How (Not) to Present Translation Suggestions

Where to Place Suggestions?

While we primarily considered using ghost text to inject MT or TM matches, participants also suggest placing them in a dedicated space, such as a toolbar on the side Figure A.5 or underneath the sentence that is being written in the target document (Figure 4.3, purple ink).

How Often Should Suggestions Be Updated?

P5 mentions that sometimes it would be better to have 'stagnant' suggestions. The participant feels that suggestions change too frequently in CAT tools with predictive typing, and that this 'makes [the system] look very unsure of itself, which is not reassuring.' P1 sometimes escapes the CAT tool when they need to focus:

> There are times when I copy a paragraph from [the CAT tool] and I put it on a Word document and I fiddle with it offline. I really need to take time and take a step back and look at it without the translation memory pushing me.

Suggestions could be updated less frequently, or they could be made optional altogether, in the sense that they only appear when triggered by translators (pull versus push). While this could give translators more freedom and be less distracting – features that participants like about working without CAT tools (Section 4.3.1) – it could, at the same time, satisfy

their demand for inspiration when they are stuck (Section 4.3.2).

How Many Suggestions?

Previous research shows that translators tend to ignore all but the first translations suggested by the CAT tool when there are multiple suggestions (Green et al., 2014a), and that showing multiple suggestions slows them down (Koehn, 2009). To our surprise, several participants told us that they consult multiple web resources to obtain translation variants, and then chose the best match for the given context. Integrating these resources into CAT tools was a key demand in the ideation phase of our survey, and two participants explicitly asked for n- rather than 1-best suggestions (Section 4.3.2). It would seem that translators require translation variants for inspiration rather than saving time, a factor that might have been underestimated (or not measured) in previous studies.

Should Markup Be Auto-projected?

One participant says that the CAT tool of their dreams would feature rich text editing (P8). While we did not discuss this in interviews, it is yet to be explored whether participants would prefer (error-prone) auto-projection of markup (tags) into the target text. The two alternatives we see are having translators reproduce markup manually through rich text editing tools, or transfer it semi-automatically by selecting the corresponding markup in the source text.

Technical Considerations

Injecting translation suggestions and projecting markup is challenging from a technical perspective as it requires accurate real-time sentence segmentation and alignment. A detailed discussion is beyond the scope of this chapter.

Considerations for Follow-up Studies

While we were worried about asking professional translators to donate some of their time for our study, we noticed that participants were grateful for us to ask for their opinion. After the interview, P8 mentioned that 'in fact you guys are working for us, to allow us to have a better workspace [...]. I'm very proud to be helpful.'

Interviewing participants remotely worked well overall. In two cases, a misunderstanding due to different time zones required us to reschedule the interview; we will use a scheduling platform going forward. Furthermore, P6 suggested sharing the questions ahead of the interview, giving them more time to think about particular topics and concepts. While we consciously opted for spontaneous feedback in this survey, we will consider this suggestion for follow-up studies.

4.5 Summary

In semi-structured interviews with 8 professional translators, we find that visual context, together with knowledge acquisition and translation alternatives, is an area in which translators see room for improvement in CAT tools. Participants see a potential for document-level editing to improve translation quality and usability, but voice concerns about orientation and question the suitability for text types other than 'actual' documents, such as lists. The design recommendations we derive from this feedback include measures to help users orientate, focus on individual segments, and display translation suggestions from TMs and MT. Our findings are summarised in Table 4.3.

Translation without CAT Tools: Advantages and Disadvantages	
+ No distraction	
+ Freedom	
+ Ownership	
Translation Memories: Advantages and Disadvantages	
+ Productivity	– Productivity
+ Familiarisation	– Freedom
	– Ownership
Standalone MT: Advantages and Disadvantages	
+ Translation variants (for inspiration)	– Quality
+ Productivity	
Integrated MT: Advantages and Disadvantages	
+ Productivity	– Quality
+ Inspiration	– Speed of adaptation
+ Adaptation	– Speed of response (latency)
	– Presence when not suitable
Ideation: Room for Improvement (See Features in Table 4.2)	
· Knowledge acquisition	
· Translation variants (for inspiration)	
· Visual context	
Document-level CAT tools (concept testing): Opportunities and Threats	
+ Quality	– Orientation
+ Usability	– Focus
+ Stand out from other CAT tools	– Text types (lists, etc.)
+ Better reviewing	

Table 4.3: Summary of findings.

Chapter 5

Impact of Text Presentation on Translator Performance

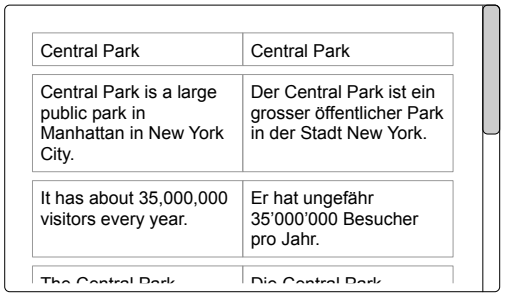
Research into CAT tool adoption among professional translators shows that poor usability is a major reason for resistance (Section 4.1). The sentence-by-sentence presentation of texts, for example, was criticised by translators for creating an 'obstructed view of the text, which in turn disrupts the [translation] workflow' (O'Brien et al., 2017). However, the impact of poor usability on translator performance has hardly been tested empirically (Section 5.1.2). Since the motivation for using CAT tools is primarily economic – saving time by post-editing translation suggestions rather than translating from scratch (Section 2.3) – the design decisions made in these tools are unlikely to change until measurements show that they slow translators down or cause them to make mistakes.

In this chapter, we test the impact of text presentation on translator performance. Our motivation is two-fold: controlled experiments show that text presentation affects reading performance (Hornbæk and Frøkjær, 2001; Yu and Miller, 2010), and that access to linguistic context affects judgement of translation quality (Chapter 3); qualitative research finds that text presentation in CAT tools is irritating (Section 4.1), and that some translators think that working with continuous rather than segmented text would help solve some 'really hard trouble' in their daily work (Section 4.3.3). We hypothesise that the empirical findings from experiments on reading and quality evaluation, two inherent activities when working with CAT tools, will carry over to computer-aided translation and substantiate concerns expressed by professional translators.

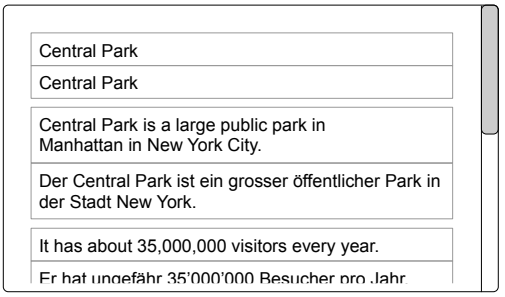
Our investigation is focussed on two aspects of text presentation: segmentation and orientation. Most widely used CAT tools[1] segment texts into sentences and present them in a side-by-side, spreadsheet like view (Figure 5.1a). Sentence segmentation is a natural choice from a technical perspective because TMs and MT operate on the level of sentences,[2] but translators consider document-level discourse. When sentences are placed

[1] Examples include Across, MemoQ, and Trados Studio (Schneider et al., 2018).

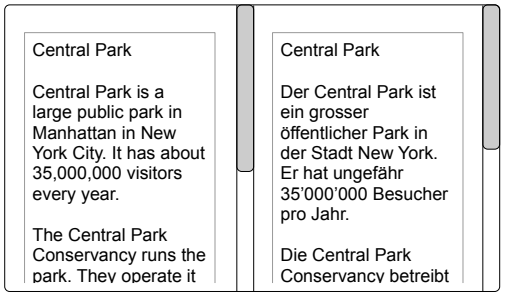
[2] As described in Section 2.3.2, TMs can be configured to operate at the level of paragraphs, but since retrieval rates are lower, sentence-level segmentation is more common. At the time of writing, document-



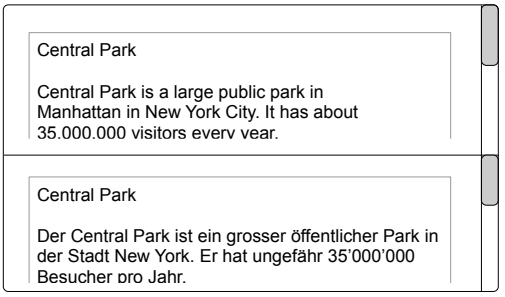
(a) Sentence, left–right (SL)



(b) Sentence, top–bottom (ST)



(c) Document, left–right (DL)



(d) Document, top–bottom (DT)

Figure 5.1: User interface configurations evaluated in this chapter. We test the efficacy of sentence segmentation vs. full document presentation and top–bottom vs. left–right orientation.

in separate boxes, inter-sentential references, such as a pronoun and its antecedent, are placed further apart, so a UI that presents continuous text may be more suitable for spotting errors related to textual cohesion (compare Figures 5.1a and 5.1c). Similarly, the distance between a word in the source and its suggested translation in the target text is larger when sentences are shown side-by-side (Figure 5.1a) compared to a top-and-bottom configuration (Figure 5.1b). Green et al. (2014a) conjecture that the latter would reduce gaze shift, the time it takes translators to realign their line of sight to the relevant segment, and we assume that a UI that eases visual orientation will lead to faster and more accurate translation.

Measuring translator speed and accuracy in controlled translation experiments is challenging: a subject cannot be exposed to the same translation in different conditions due to repetition priming, and translation quality is difficult to define and measure (House, 2013; Green et al., 2013). To control for confounding variables, we focus on specific activities that can be relevant when working with CAT tools – text reproduction, within-sentence error identification, and document-level revision – and design our experimental tasks such that accuracy[3] can be measured with minimal ambiguity. In the revision task, for example, we insert errors into human translations that are unambiguously wrong, and measure whether and how quickly subjects correct these errors within the different UIs.

We review related work, and previous studies that used similar means of decomposition in particular, in Section 5.1, and detail our experimental design in Section 5.2. Results are presented in Section 5.3: we find significant evidence that a top-and-bottom arrangement of sentences enables faster text reproduction and error identification. For revision, on the other hand, our results suggest that a side-by-side presentation of full documents results in the highest accuracy and time efficiency. Implications for best practices in designing CAT tools and limitations are discussed in Section 5.4.

5.1 Background

Our interaction with computers, machines that carry out mathematical operations, is mediated by UIs. When we work with graphical UIs, we tend to forget that everything we see is the result of a design process: the position, colour and size of any button and text box are not determined by chance, but by design decisions actively made by people in charge (Norman, 1988). As it is difficult to test every option with the intended audience, designers base some or all of these decisions on conventions and assumptions. A convention in the UI of a program running on the Windows operating system, for example, is to place a small red button with an 'x' symbol, whose on-click behaviour is to terminate the program, in the upper right corner. The assumption is that users will be familiar with this convention and thus know how to terminate the program, but if there are no conventions

level MT is not available commercially, but shows promising results in industrial research (e.g., Junczys-Dowmunt, 2019). Implications are discussed in Section 5.4.

[3]Throughout this article, we use the term accuracy rather than quality to emphasise that we focus on specific linguistic phenomena that are categorisable as correct or incorrect with no or minimal ambiguity.

or if conventions are considered suboptimal, metaphors are a powerful tool for designers. For instance, the adoption of personal computers soared after the metaphor Computer is a Desktop replaced the conception that a computer is a programming environment (Saffer, 2005).

5.1.1 Text and Document Visualisation

Text editing has been a fundamental task supported by modern computers since their inception (Engelbart and English, 1968). Early text editors were referred to as line editors because, mostly due to hardware constraints, users were required to select, manipulate, and then display individual lines of a document in separate steps; manipulation and document display did not occur simultaneously. Shneiderman (1983) promoted display editors: whereas 'the one-line-at-a-time view offered by line editors is like seeing the world through a narrow cardboard tube', a display editor always shows a document in its final form and 'enables viewing each sentence in context and simplifies reading and scanning'. The visualisation of full documents has a direct impact on productivity: display editors were shown to double text editing speed compared to line editors (Roberts, 1980; Roberts and Moran, 1982).

While the 'what you see is what you get' (WYSIWYG) principle seen in display editors has long become the standard in word processing software we use in our everyday life, adjustments in text presentation have further improved UIs for text editing. Hornbæk and Frøkjær (2001) investigate if two alternatives to a regular (referred to as linear) UI improve reading speed and comprehension of electronic documents: a fisheye UI that shrinks certain parts of the document below readable size, which can be made readable by clicking on them; and an overview+detail UI that displays a miniaturised version of the document in a sidebar (the overview pane) that can be clicked to quickly move the main pane (referred to as the detail pane) to a desired section. A controlled experiment with 20 subjects finds that while the fisheye UI improves reading speed, the overview+detail UI improves reading comprehension and achieves the highest satisfaction among subjects, ten of which 'mention the overview of the documents structure and titles as an important reason' (ibid.). While we do not use an overview pane in our experimental interfaces, we observe that much of a document's structure is lost in the sentence-level UIs of widely used CAT tools (Figure 5.3a), while a UI that presents unsegmented text retains structural cues such as titles and paragraphs (e.g., Figure 5.3c). Yu and Miller's (2010) Jenga format is a compromise between the two: it separates paragraphs into sentences, but, in contrast to CAT tools, only adds vertical space while the horizontal position of each sentence remains unchanged. A user study with 30 subjects finds that presenting texts in this format significantly enhances web page readability (ibid.).

5.1.2 Text and Document Visualisation in CAT Tools

Context is also vital to produce high-quality translations, yet this context is often narrow in CAT tools. One reason is that, in contrast to regular word processors, the UI of a CAT tool needs to accommodate two documents – the (generally uneditable) source document and its translation, the target document – and additional panes to display translation suggestions. All of these elements compete for space on the translator's screen, and while the size of these elements is typically configurable, showing more translation suggestions at once, for example, will necessarily decrease the number of source and target sentences that can be shown without scrolling. Another reason is that the source and target documents are rendered as a table where each sentence is placed in a separate cell. If a sentence does not use the full width of a cell, or if either the source or the target sentence uses more lines than its counterpart, some of the space remains blank, which further limits the number of sentences that can be viewed without scrolling. A UI that shows continuous text can accommodate more text – and thus more context around the sentence being translated (compare Figures 5.1a and 5.1c).

The fact that widely-used CAT tools visualise documents as tables rather than continuous text implies a common motivation among manufacturers, and the question that arises is whether this motivation is rooted in ergonomic considerations. From a user's perspective, the Document is a Table metaphor seems less intuitive than the Document is a Series of Pages metaphor used in applications like Microsoft Word, which, despite not offering translation functionality, is used for MT post-editing by 38 % of professional translators (Moorkens and O'Brien, 2017). Translation process research finds that the sentence-by-sentence presentation in CAT tools 'creates an unnaturally strong focus on the sentence' that reduces the number of changes made to sentence structure in translations (Dragsted, 2006), and ethnographic studies as well as surveys with professional translators conclude that the segmented view of documents is problematic (e.g., LeBlanc, 2013; O'Brien et al., 2017). Some of our interviewees, on the other hand, highlight positive aspects: P3 and P5 describe sentence segmentation as helpful to focus and keep track of where they are, and similarly, P1 recalls that having to jump back and forth with their eyes between the source and target document felt tiring before using CAT tools (Section 4.3.3).

While we have been unable to find published information that motivates the use of sentence segmentation by commercial CAT tool providers, a review of academic research suggests that design choices on document visualisation are not based on empirical investigation. Kay (1980) suggests incorporating simple translation functionality into word processors. He theorises that this editor would be 'divided into two windows. The text to be translated appears in the upper window and the translation will be composed in the bottom one'. This suggestion – a document-level UI with top–bottom orientation (Figure 5.2) – was later implemented in TransType: Langlais et al. (2001) 'tried to display the text and its translation side by side but *it seems* that a synchronized display of the original text and its translation one over the other is better' (emphasis added). Green et al. (2014a) present a UI that uses a top-and-bottom arrangement of source and target sentences (Fig-

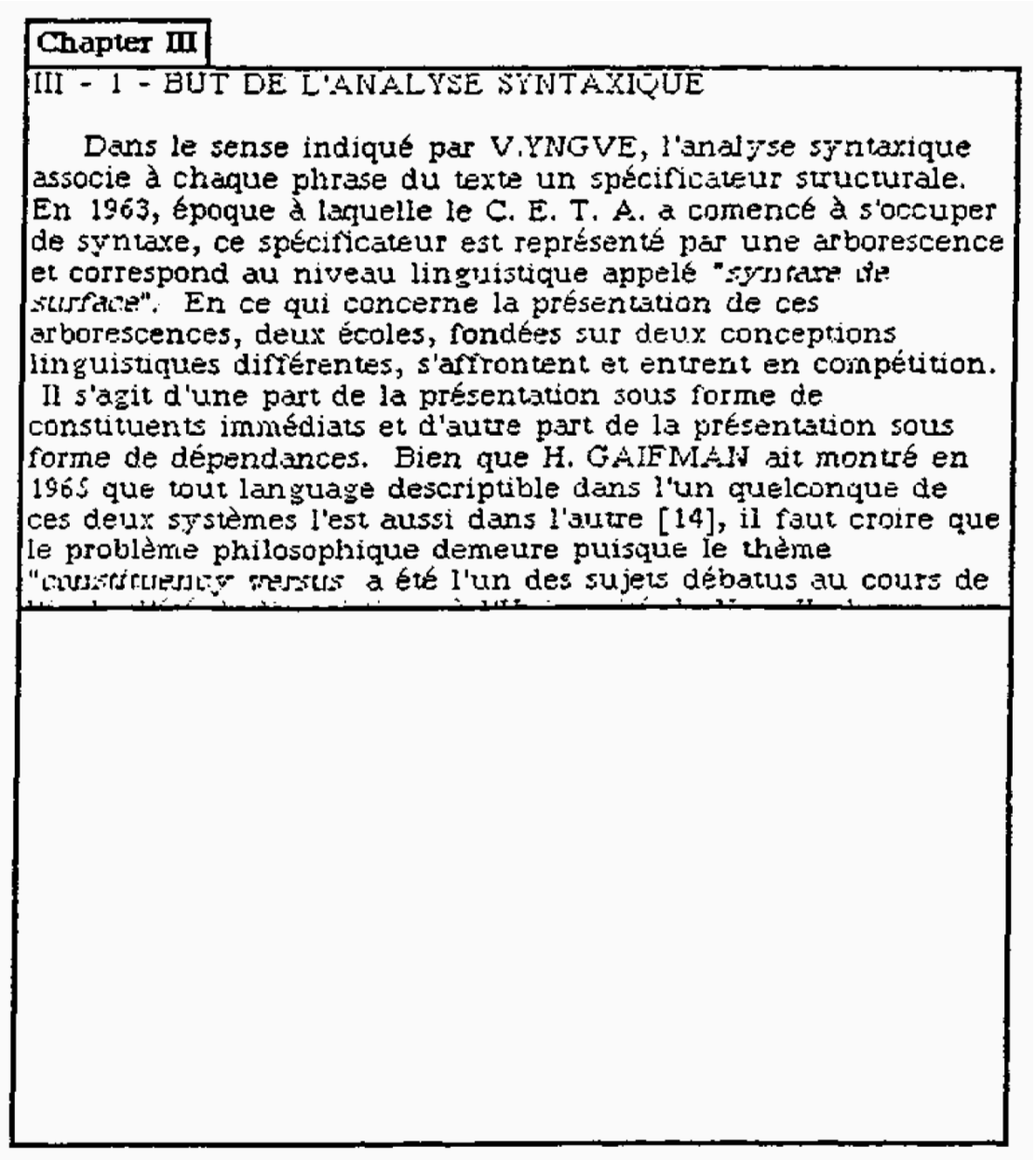


Figure 5.2: Kay's (1980) theorised collaborative man–machine system for translation. The document-level UI presents continuous source (top) and target text (bottom), akin to the DT interface we test in our experiment (Figure 5.1d).

ure 5.1b). The authors ground this design choice in the observation that translators spend up to 20 % of their time reading when translating a document (Carl, 2010), and argue that their

> UI is based on a single-column layout so that the text appears as it would in a document. Sentences are offset from one another primarily because current MT systems process input at the sentence-level. We interleave target-text typing boxes with the source input to minimize gaze shift between source and target. Contrast this with a two-column layout in which the source and target focus positions are nearly always separated by the width of a column.

Green et al.'s (2014a) investigation is focused on interaction features and does not assess the impact of top–bottom orientation. The CAT tool prototype evaluated by Coppers et al. (2018) also uses the design proposed by Green et al. (2014a), but the authors do not evaluate it against a UI that uses left–right orientation, a gap we fill with the experiment presented in this chapter.

5.1.3 Understanding Translator Performance

Our aim is to assess the impact of text presentation on translator performance, and a fundamental question in translation experiments is how translator performance should be defined and measured. Some experimental designs maximise external validity: they measure temporal effort and/or the quality of products under realistic working conditions, the goal being that results will reflect the 'truth in real life' (e.g., Federico et al., 2012). Apart from resource-related challenges such as high cost (e.g., because subjects should be professional translators rather than students), such experimental designs limit control of extraneous variables (e.g., because the user-defined settings in a CAT tool cannot be standardised when subjects use their own workstation) and insights into why a particular result was obtained (e.g., whether slower subjects spend more time on reading or writing). Moreover, realistic working conditions may not be achievable in the context of fundamental research not only because subjects will necessarily be unfamiliar with the research prototypes to be tested, but also because prototypes will typically not implement all of the functionality available in commercial products.

For some or all of these reasons, other experimental designs in translation research maximise internal validity. In ensuring that results will reflect 'the truth in the study', such designs may involve resources and procedures that deviate from realistic working conditions for better control (e.g., control for screen size by having all subjects work on a standardised workstation in a lab) or finer-grained measurements (e.g., how much time subjects spend reading and writing). The investigation of Krings (1994, 2001),[4] for example, aims at gaining an understanding of how translation processes change as translators post-edit MT rather than translate from scratch.[5] Krings asks subjects to Think Aloud

[4]We reference page numbers in the English translation of Krings's (1994) habilitation thesis (Krings, 2001) due to better availability and accessibility.

[5]Even if his study is best known for the finding that the temporal effort for translation from scratch and

(Ericsson and Simon, 1984) as, using pen and paper, they translate or post-edit, the latter without access to the source text in one task of the experiment. Although very different from a translator's regular working conditions, this setup allows the author to elaborate and quantify the relative distribution of sub-processes, such as target text monitoring or writing, in translation from scratch and post-editing. The use of Think Aloud protocols is known to impact translation speed (Jakobsen, 2003), and other data collection methods such as key-logging and eye-tracking likewise pose challenges to external validity (e.g., O'Brien, 2009); but while results such as time measurements from such experiments may not be directly transferable to real-life situations, conclusions drawn from comparing measurements between experimental conditions may well be. With respect to Krings (2001): while it may not hold that 42.5 % and 43.5 % of the processes in translation from scratch and post-editing, respectively, relate to target text production under normal working conditions (ibid., p. 314), it is plausible that the difference will also be small under normal working conditions since the aggravating circumstances were the same in both tasks of the experiment.

Since no commercial CAT tool implements all of the UIs we test in our experiment,[6] the use of prototypes is inevitable, and our goal cannot be to predict how text presentation will affect translation under real-life working conditions with commercial CAT tools that provide many more functions than these prototypes (Section 5.2.2). Instead, we are interested if, and to what extent, the different UIs impact the speed and accuracy of professional translators when all but segmentation and orientation – such as font size, spacing, etc. – stays exactly the same. Our experimental design choices are guided by two principles aimed at maximising internal validity. First, we do not categorise translation processes, but define specific tasks for particular processes (Section 5.2.1). To assess how the UIs affect reading, for instance, we do not ask subjects to translate a text and then try to identify in which parts of the translation sessions subjects were reading; we define a specific reading task (Scan). Second, we define response variables that are measurable with no or minimal ambiguity (Sennrich, 2017). To assess if the UIs impact the number of typing errors, for example, we do not look for typing errors in freely written translations; we ask translators to reproduce a given text (Copy) so we can calculate the number of typing errors (the Levenshtein distance) exactly. The specifics of our experimental design are detailed in the next section.

5.2 Experimental Methods

We conduct a controlled experiment to empirically test the impact of text presentation on translator performance. We use a mixed factorial design and measure time and accuracy in three experimental tasks. The independent variables (factors) are UI segmentation (S: sentence, D: document), UI orientation (L: left–right, T: top–bottom), and texts. Seg-

MT post-editing (in 1994) is roughly the same (Krings, 2001, p. 552), which, as such, can also be tested with an extrinsic design (Federico et al., 2012).

[6] On the contrary, we are not aware of any CAT tool in wide use that implements a document-level UI.

mentation and orientation are within-subjects factors, while text is a between-subjects factor: subjects see all factor levels in each task, but not all combinations since processing the same text twice induces repetition priming (Francis and Sáenz, 2007).

5.2.1 Tasks

We define three experimental tasks: text reproduction (Copy), error identification (Scan), and revision (Revise). We measure speed and accuracy in each task, and minimise ambiguity in the latter by means of contrastive evaluation (Sennrich, 2017) in the Scan and Revise tasks: experimental items are manipulated by inserting an artificial error, and the binary response variable encodes whether or not subjects identify (Scan) or correct (Revise) the error.

Text Reproduction (Copy)

In cases where no TM or MT suggestions are available, translators read source text and produce target text. These activities are interleaved (Ruiz et al., 2008; Dragsted, 2010), and we want to assess how interleaved reading and writing is affected by text presentation. However, this process involves comprehension (ibid.), and to avoid that subjects will spend time on source comprehension and target generation problems – which are difficult to control for as they will vary among participants and texts – we ask subjects to copy source text into the target text box(es) of our experimental UIs. As such, this task can be related to situations where translators actually reproduce source text in a translation (e.g., a list of product names), or the technical effort in regular translation (i.e., overall effort minus time spent on problem solving). We enforce manual typing by suppressing the use of copy and paste commands, and measure the time it takes subjects to type out entire texts. We calculate accuracy as the number of mistyped characters per text (Levenshtein distance).

Error Identification (Scan)

Translators increasingly work with suggestions from TMs or MT systems (do Carmo and Moorkens, 2020), which involves target text comprehension: translators scan translation suggestions to decide whether they can be used as-is or need adjustment. Special care must be taken when working with suggestions from neural MT systems as they may read fluently, but contain omitted, added, or mistranslated words (Castilho et al., 2017b, 2018b). In the Scan task, we are interested in whether text presentation impacts the speed and accuracy with which translators can identify such mistakes, which we simulate for better measurability (Section 5.1.3): we either repeat (Addition) or delete (Omission) word sequences in translations produced by human professionals, insert nonsensical sentences (Wrong Meaning), or leave them unchanged (No Error). Examples are shown in

Table 5.1. We apply these manipulations to 10 % of randomly selected sentences (minimum: 1) in each text, roughly corresponding to the distribution of errors in English to German MT (Castilho et al., 2018b). Subjects are asked to assign each text to one of the four categories. We measure how much time they need for each judgement, and whether or not they assign the correct category.

Revision (Revise)

Translations are normally revised before dissemination, and one important aspect in revision is cohesion: making sure that the connection between sentences and/or paragraphs is appropriate (Shih, 2006). Since TMs and MT usualy operate on isolated sentences, they are prone to suggest sentences with anaphors (such as pronouns) and named entities (such as product names) that are not compatible with surrounding sentences and the document as a whole (Castilho et al., 2017a; Müller et al., 2018). In the Revise task, we test if UI segmentation and orientation impact the ability and speed of translators to correct such errors. As in the Scan task, we manipulate professional translations, and insert one error per document: a mistranslated anaphor or named entity. These errors are constructed such that they are not identifiable within single sentences, meaning subjects have to read the entire text or at least the surrounding sentences to notice them (Table 5.2). Subjects are asked to revise full documents, and are not told that we focus on anaphors and named entities specifically. We classify the revised documents they submit as correct or incorrect solely based on whether the inserted error is corrected. Any other revisions made by subjects are ignored.

5.2.2 Materials

Texts

We use German translations of English news articles in all tasks. Both the original English texts and their German translations, produced by professional translators, stem from reference data released by the organisers of the 2017 and 2018 Conference on Machine Translation (Bojar et al., 2017, 2018).[7,8] Texts are chosen at random, excluding very short and very long instances whose lengths differ by more than one standard deviation from the mean number of sentences per text in the entire collection. The selected texts contain 21.85 sentences on average (min=8, max=44, median=20.00, sd=9.91). We note that the overall quality of the German translations, which we manipulate by inserting specific errors for the Scan and Revise tasks, has been criticised (Hassan et al., 2018); we do not edit or control for errors other than the ones we insert for contrastive evaluation. This is potentially problematic for the Scan task, e.g., if a translation into which we artificially insert an addition also contains an omission produced by the original translator; in the

[7] http://data.statmt.org/wmt17/translation-task/test.tgz
[8] http://data.statmt.org/wmt18/translation-task/test.tgz

S	While sufferers are usually advised to dodge meat and dairy to soothe their symptoms, researchers at Washington University found protein's essential amino acid tryptophan helps develop immune cells that foster a tolerant gut.
T_o	Während den Betroffenen normalerweise geraten wird, Fleisch und Milchprodukte zu meiden, um ihre Symptome zu lindern, fanden Forscher an der Washingtoner Universität heraus, dass die essentielle Aminosäure Tryptophan von Proteinen dazu beiträgt, Immunzellen zu entwickeln, die einen toleranten Darm fördern.
T_m	Während den Betroffenen normalerweise geraten wird, Fleisch und Milchprodukte zu meiden, um ihre Symptome zu lindern zu lindern zu lindern, fanden Forscher an der Washingtoner Universität heraus, dass die essentielle Aminosäure Tryptophan von Proteinen dazu beiträgt, Immunzellen zu entwickeln, die einen toleranten Darm fördern.

(a) Addition

S	Patrick Roy resigned as coach and vice president of the hockey operations of the Colorado Avalanche on Thursday, citing a lack of a voice within the team's decision-making process.
T_o	Patrick Roy trat am Donnerstag als Trainer und Vice President Of Hockey Operations der Colorado Avalanche zurück und führte ein zu geringes Mitbestimmungsrecht beim Entscheidungsprozess des Teams an.
T_m	Patrick Roy trat am Donnerstag als Trainer und Vice President Of Hockey Operations zurück und führte ein zu geringes Mitbestimmungsrecht beim Entscheidungsprozess des Teams an.

(b) Omission

S	Heavy rain, flooding prompts rescues in Louisiana, Mississippi
T_o	Heftige Regenfälle, Überschwemmung gibt Anlass zu Rettungen in Louisiana, Mississippi
T_m	Öffnen Sie im zweiten Fenster das eigene Persönliche Verzeichnis.

(c) Wrong Meaning

S	The "Made in America" event was designated an official event by the White House, and would not have been covered by the Hatch Act.
T_o	Die Veranstaltung "Made in America" wurde vom Weißen Haus als offizielle Veranstaltung bezeichnet und wäre nicht vom Hatch Act abgedeckt worden.
T_m	Die Veranstaltung "Made in America" wurde vom Weißen Haus als offizielle Veranstaltung bezeichnet und wäre nicht vom Hatch Act abgedeckt worden.

(d) No Error

Table 5.1: Examples of target text manipulations in the SCAN task. Subjects see the original source S and manipulated target T_m; the original target T_o is shown here for the purpose of illustration. In the experiment, the manipulated sentences are embedded in full news articles.

S	... to an Earls Court apartment in 2014. It is on the first floor of a smart Queen Anne terrace - and it is a testament to the new design ...
T_o	... zu einem Earls Court Apartment. Es liegt im ersten Stock einer eleganten Queen Anne Terrasse - und es ist ein Beweis für das neue Design ...
T_m	... zu einem Earls Court Apartment. Sie liegt im ersten Stock einer eleganten Queen Anne Terrasse - und es ist ein Beweis für das neue Design ...

(a) Anaphor

S	Pokémon Go, a worthy hunt for health and happiness ... Within days, Pokémon Go had more users than Tinder ... But the beauty of Pokémon Go is it gets people outside doing something they enjoy ...
T_o	Pokémon Go, eine Jagd nach Gesundheit und Glück, die sich lohnt ... Innerhalb weniger Tage hatte Pokémon Go mehr Benutzer als Tinder ... Aber das Wundervolle an Pokémon Go ist, dass es die Leute dazu bringt, etwas im Freien zu tun ...
T_m	Pokémon Go, eine Jagd nach Gesundheit und Glück, die sich lohnt ... Innerhalb weniger Tage hatte Pokémon Go mehr Benutzer als Tinder ... Aber das Wundervolle an Pokémon Gehen ist, dass es die Leute dazu bringt, etwas im Freien zu tun ...

(b) Named Entity

Table 5.2: Examples of target text manipulations in the Revise task. Subjects see the original source S and manipulated target T_m; the original target T_o is shown here for the purpose of illustration. In the experiment, the manipulated passages are embedded in full news articles.

Revise task, additional errors cannot influence our measurements since we ignore edits other than those made to our manipulations, and in the Copy task, subjects only work with the source texts (see above).

User Interfaces (UIs)

We test four experimental UIs that differ in text presentation, resulting from crossing the levels of two experimental factors: segmentation and orientation. Sentence-level UIs show sentences in individual text boxes, akin to most CAT tools currently used by professional translators; document-level UIs show the full text in a single text box, as seen in regular word processors. Interfaces with left–right orientation place target text to the right of the corresponding source text, while target text is placed underneath the corresponding source text in UIs with top–bottom orientation. A screenshot of each UI is shown in Figure 5.3.

All UIs use a fixed-width content pane of 1024 by 576 pixels, a 16:9 ratio (the white area in Figures 5.3a–d). Working with texts exceeding the height of this pane requires vertical scrolling. The scrolling behaviour differs between sentence- and document-level UIs in that the former use a single scroll bar (e.g., Figure 5.3b), while document-level UIs use individual scroll bars for the source and target text boxes (e.g., Figure 5.3d). This behaviour is criticised by a number of subjects in our post-experiment survey, as discussed further in Section 5.4.3.

Typographic choices are based on design guidelines for on-screen readability (Rello et al., 2016; Miniukovich et al., 2017). We use a 12 pt sans-serif font (Arial) to typeset source and target text in dark grey and black colour, respectively, with 150 % line spacing, left justification, and ragged right edge.

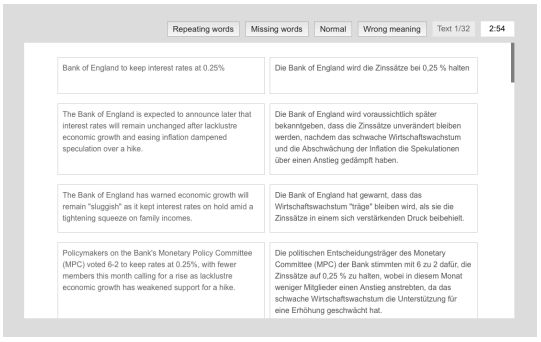
5.2.3 Subjects

We recruit 20 professional English to German translators (S1–20) from a multinational language services provider, excluding individuals who have participated in our interviews (I1–8, Chapter 4).

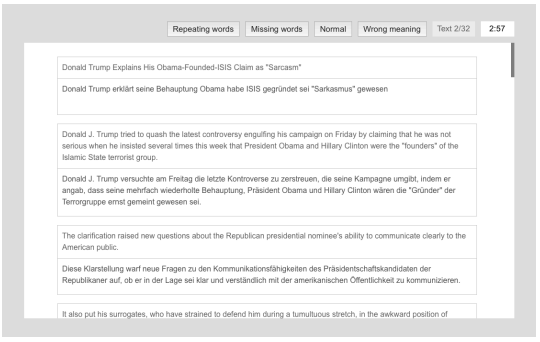
We pay each translator $ 245.00 for completing the entire experiment. With an average duration of 7.55 hours, this corresponds to an hourly rate of $ 32.45, close to the industry average of $ 35.57.[9]

Subjects complete a pre-experiment survey in which we elicit information on their personal background. On average, subjects have 10.70 years of professional translation experience (min=1, max=28, median=6.50, sd=9.12). 12 out of 20 subjects have a university degree in translation, and 10 have some background in information technology,

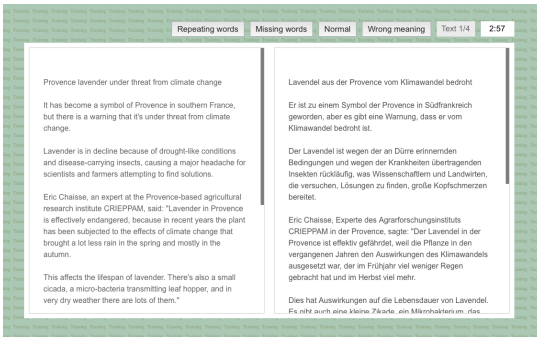
[9] According to rates reported by freelance translators and translation companies on ProZ, a large online translation community: `https://search.proz.com/employers/rates?source_lang=eng&target_lang=deu&disc_spec_id=&currency=usd`.



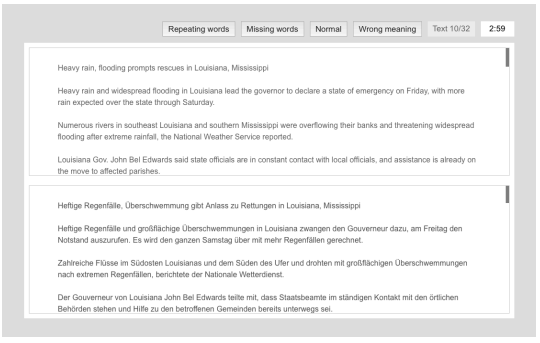
(a) Sentence, left–right (SL)



(b) Sentence, top–bottom (ST)



(c) Document, left–right (DL)



(d) Document, top–bottom (DT)

Figure 5.3: Experimental UIs. Subjects process at least three texts per UI in each task, and one text each in the training phase. The visual cue for the training phase is shown in sub-figure c.

	None	Courses	Bachelor's	Master's	PhD
Translation	5	3	7	4	1
Information Technology	10	9	0	1	0

(a) Education (highest degree)

	Regular	Sometimes	Never
Translation Memory	15	4	1
Terminology Management	7	9	4
Machine Translation	5	10	5
Quality Assurance	4	8	8

(b) Use of translation technology

Table 5.3: Background information by number of subjects (pre-experimental survey).

mostly from attending specialised courses, seminars, or workshops (Table 5.3a). This distribution is very similar to that of a larger group of professional translators surveyed by Zaretskaya (2015), whereas the percentage of regular or occasional users of translation technology is higher among our subjects (Table 5.3b).

5.2.4 Procedure

As most professional freelance translators work from home (Ehrensberger-Dow et al., 2016), we opt for a browser-based remote experiment. Subjects complete the experiment using their own computer from a workplace of choice. We block access to the experiment with unsupported browsers, mobile devices, or screens with a resolution that is not large enough to accommodate the entire content pane (see above). We send out general instructions via email, and have a 15-minute introductory call to clarify questions with each subject.

We randomly choose 12, 32, and 12 texts for the Copy, Scan, and Revise tasks, respectively, plus four texts per task for training. We determine the number of items in a pilot run with two professional translators who do not participate in the final experiment. We include more items in Scan so as to collect sufficient responses for each error type (Addition, Omission, Wrong Meaning, and No Error). Each subject is presented with the same texts in each task, but not in the same UIs; to control for order effects, we use a Latin square assignment of texts to UIs, and counterbalance the order of tasks among subjects.

Subjects complete the entire experiment in a single workday. Each task starts with a training phase: subjects read through the task instructions and then complete the four training items, one in each UI (in random order). We use a visual cue to distinguish training from experimental items (Figure 5.3c), and subjects can repeat each training phase as often as they wish. They can take breaks between tasks, but must complete all items within a task without breaks and under time pressure: we display an idle timer that is reset upon any keyboard or mouse activity, and trigger automatic submission of the current item if no such activity is recorded within three minutes. Time pressure is common in professional translation (Ehrensberger-Dow et al., 2016) and may increase cognitive function (Campbell, 1999), but we avoid a fixed deadline to account for per-subject and per-item variation.

Subjects complete a survey before the experiment, one after each task, and one after the experiment. They can optionally leave free-form feedback.

5.2.5 Data Analysis

Our response variables are time and accuracy. We measure total wallclock time per item, which we log-transform for statistical analysis as our response time measurements follow a log-normal distribution. The coding of accuracy depends on the task, as described in the following section. We report speed in words per hour for Copy and Revise, and

in seconds per item for SCAN since subjects can make correct judgements without reading through the entire text, thus biasing normalisation by length. We define a word as 5 source text characters, including spaces (Arif and Stuerzlinger, 2009).

We fit linear and logistic mixed-effects models to our measurements for continuous and categorical response variables, respectively, using the `lme4` package in R (Bates et al., 2015). We use a random effects structure with random intercepts for subjects and texts in all of our models (Green et al., 2013), and apply mild a-priori screening in combination with model criticism to detect outliers (Baayen and Milin, 2010). We check for deviations from homoscedasticity or normality by visual inspection of residual plots and Shapiro-Wilk tests in linear models, and inspect logistic models for overdispersion problems and high error rates (Gelman and Hill, 2007).

5.3 Experimental Results

5.3.1 Text Reproduction (COPY)

Out of the 240 responses in COPY, we exclude 2 responses triggered by automatic submission due to subject inactivity (no keyboard or mouse input) for three minutes (Section 5.2.4).

Speed

A-priori screening removes 8 responses with response times deviating by more than 2.5 standard deviations from the per-text (5) and per-subject (3) medians. We fit a linear mixed-effects model for log-transformed response time with fixed effects for segmentation and orientation, and remove 4 overly influential outliers with large residuals through model criticism.

Likelihood ratio tests find significant effects for both segmentation ($\chi^2(1)$=14.58, p<.001) and orientation ($\chi^2(1)$=5.66, p<.05): sentence-level is faster than document-level segmentation, and top–bottom is faster than left–right orientation. A model with an interaction term for segmentation and orientation does not improve model selection scores (i.e., the Akaike (AIC) and Bayesian (BIC) information criteria), and the interaction is not significant ($\chi^2(1)$=0.94, p=.33).

Accuracy

We remove 6 responses that contain between 328 and 2230 mistyped characters, more than 2.5 standard deviations from the global mean, which indicates rashness or unintentional submission before completion. Another 6 observations with large residuals are removed through model criticism.

Task	Copy		Scan		Revise: Named Entity		Revise: Anaphor	
Response Variable	Speed	Accuracy	Speed	Accuracy	Speed	Accuracy	Speed	Accuracy
Unit	words/h	# typos	s/item	% correct	words/h	% correct	words/h	% correct
Sample Size	226	226	413	636	63	99	57	119
Mean Response								
Sentence, left–right (SL)	2,523.16	9.49	114.31	66.46	5,037.84	61.54	4,693.47	48.27
Sentence, top–bottom (ST)	2,639.13	8.07	123.80	69.38	4,323.27	76.00	4,511.23	37.93
Document, left–right (DL)	2,463.37	9.19	153.98	65.19	4,786.27	70.83	5,350.81	56.67
Document, top–bottom (DT)	2,459.79	9.28	145.83	68.75	4,365.75	66.67	4,778.32	58.07
Effects								
Segmentation	•••		••					○
Orientation	•				••			
Experience with MT	*n/a*	*n/a*	••	•••	*n/a*	•	*n/a*	
Error Type	*n/a*	*n/a*	•••	•••	*n/a*	*n/a*	*n/a*	*n/a*

Table 5.4: Summary of experimental results. Significance levels are denoted by ○ $p<.1$, • $p<.05$, •• $p<.01$, and ••• $p<.001$.

Predicting the number of mistyped characters from segmentation and orientation results in heteroscedastic residuals, so we apply a sqrt-transformation to the dependent variable; a log-transformation is not applicable since 0 – no typing errors at all – is a valid response. Likelihood ratio tests find no significant effects for segmentation ($\chi^2(1)$=0.59, p=0.44) and orientation ($\chi^2(1)$=2.08, p=0.15), even with more complex models that include additional predictors such as translator experience or familiarity with translation technology.

5.3.2 Error Identification (SCAN)

Speed

We are interested in whether text presentation influences the time needed to make correct judgements of target text quality, so we consider correctly labelled (433 out of 640) responses for the SCAN time model. We remove 4 responses with response times below 1 second, and fit a linear mixed-effects model with fixed effects for segmentation and orientation. 16 responses are removed through model criticism. A stepwise variable selection procedure results in a model with better AIC (930.9 vs. 956.6) and BIC (975.2 vs. 980.8) scores, which includes two additional fixed effects: error type and experience with MT. The latter is elicited in the pre-experiment survey, where we ask subjects if they have used MT 'regularly', 'sometimes', or 'never' (Section 5.2.3).

Subjects are significantly faster with sentence-level segmentation ($\chi^2(1)$=7.34, p<.01). Likelihood ratio tests do not find a significant effect for orientation ($\chi^2(1)$=0.11, p=.74), but for error type ($\chi^2(3)$=33.10, p<.001) and experience with MT ($\chi^2(2)$=7.71, p<.05). Subjects detect translations with Wrong Meaning quickly, and need much longer to identify translations that are not manipulated (No Error). This is not surprising since making sure that a translation contains no errors requires that it be read to the end, while subjects can stop reading as soon as they find an error in a manipulated translation. In terms of MT, subjects who have used the technology regularly are the slowest, but also the most accurate.

Accuracy

We use a binary coding of 1 (when subjects assign the correct class) and 0 (otherwise) for the response variable in SCAN accuracy. After again removing the 4 outliers with response times below 1 second, we fit a logistic mixed-effects model using the same mixed-effects structure as in the time model.

Likelihood ratio tests find no significant effects for segmentation ($\chi^2(1)$=0.11, p=.74) and orientation ($\chi^2(1)$=1.41, p=.24), but for error type ($\chi^2(3)$=60.67, p<.001) and experience with MT ($\chi^2(3)$=12.98, p<.01). Subjects label 89.2 % of translations with wrong meaning correctly, more so than translations with missing words (61.6 %), repeated words (62.2 %), and translations that contain no error (56.9 %). This indicates

that translations we have not manipulated contain errors that we do not control for (Section 5.2.2). As for experience with MT, 80.0 % of responses produced by subjects who have regularly used the technology are correct, more than those of subjects who have sometimes (65.7 %) or never (58.2 %) worked with MT.

5.3.3 Revision (REVISE)

We build separate models for texts manipulated with a wrong named entity and a wrong anaphor, and remove one response each where no keyboard or mouse activity was recorded for more than three minutes (Section 5.2.4).

Speed

As in SCAN, we model response time for texts which subjects revised correctly. A-priori screening removes 5 responses (2 with a manipulated named entity, 3 with a manipulated anaphor) with response times deviating by more than 2.5 standard deviations from the per-text median, leaving a total of 63 and 57 correct responses for texts with a manipulated named entity and anaphora, respectively.

We find a significant effect for orientation with texts containing a manipulated named entity ($\chi^2(1)$=6.30, p<.05), which subjects revise faster with left–right UIs. Effects for segmentation with these texts ($\chi^2(1)$=0.08, p=.77) as well as both segmentation ($\chi^2(1)$=0.01, p=.91) and orientation ($\chi^2(1)$=0.15, p=.70) with texts containing a manipulated anaphor are not significant.

Accuracy

We use a binary coding of 1 (when subjects correct the error we deliberately inserted) or 0 (when they do not) in the response variable for accuracy. We fit a logistic mixed-effects model each to the responses from texts with a manipulated named entity and anaphor, excluding one text with a named entity that none of the subjects revise correctly (20 responses). We include experience with MT as a fixed effect in addition to segmentation and orientation, which improves model selection scores for the named entity model and leads to lower error rates (Gelman and Hill, 2007) in both models.

For texts with a manipulated named entity, effects for segmentation ($\chi^2(1)$=0.00, p=.94) and orientation ($\chi^2(1)$=0.61, p=.44) are not significant, but revision by regular users of machine translation (experience with MT) is significantly less accurate ($\chi^2(2)$=6.55, p<.05).

For texts with a manipulated anaphor, likelihood-ratio tests find a near-significant effect of segmentation ($\chi^2(1)$=5.53, p=.06), where mean accuracy is higher with document-level UIs. Effects for orientation ($\chi^2(1)$=0.26, p=.61) and experience with MT ($\chi^2(2)$=1.28, p=.53) are not significant.

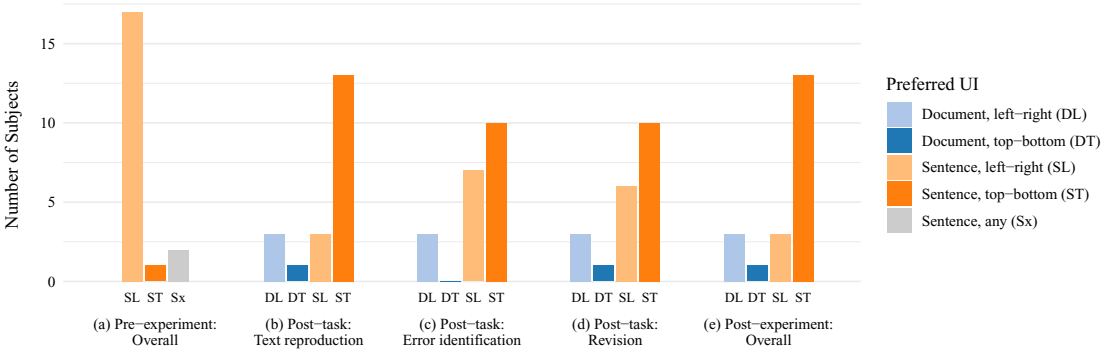


Figure 5.4: UI Preferences. The majority of subjects are used to sentence-level UIs with left–right orientation (a), but prefer top–bottom orientation in the experiment (e).

5.3.4 UI Preference

We ask subjects about their preferred orientation (left–right or top–bottom) in the CAT tools they usually work with in the pre-experiment survey, and contrast this with feedback on the experimental UIs in three post-task surveys (one each after Copy, Scan, and Revise) and a post-experiment survey. As shown in Figure 5.4, most subjects (85 %) use left–right orientation in their daily work. In the experiment, however, the majority (65 %) prefer the sentence-level UI with top–bottom orientation.

5.4 Discussion and Design Implications

The triangulation of preliminary feedback from potential users (I1–8, Chapter 4), our empirical results, and feedback from experimental subjects (S1–20) yields new design principles for text presentation in CAT tools. We also discuss the limitations of our study in this section, and outline directions for future work.

5.4.1 Segmentation

CAT tools in wide use visually separate the sentences in a document. Our results suggest that this is helpful for sentence-level tasks: reproducing text (Copy) and identifying errors within sentences (Scan) are significantly faster in UIs with sentence-by-sentence presentation (Table 5.4). In post-experiment feedback, several subjects note that it is 'easier to work with a text when it is separated into segments' (S4), notably because it 'eliminates [the] need for scrolling and paragraphing' (S1).

For tasks that require super-sentential context, on the other hand, segment-by-segment presentation provides no advantage over a display of continuous text. On the contrary, anaphoric relations are revised more accurately in document-level UIs, and while the

difference to sentence-level UIs is not significant (p=.06), the effect size is considerable: 58.07 % in DT vs. 37.93 % in ST, the latter performing best in Copy and Scan (Table 5.4).

In this light, the characterisation of sentence-by-sentence presentation as 'unnatural' (Dragsted, 2006) or 'irritating' (O'Brien et al., 2017) in other translation research and the largely positive feedback from our subjects are not necessarily conflicting: the suitability of sentence segmentation depends on the task, and the problem may be that it cannot be turned off when it is not suitable. Since the translation activities that motivate our experimental tasks are interleaved in practice (Ruiz et al., 2008; Dragsted, 2010), letting translators switch between a segmented and continuous view of the document they are translating may enable them to focus on local context and consider global context when needed.

This is supported by feedback from our interviewees, who figure that 'you need segmentation sometimes to just better focus' (I5), and believe that combining sentence-by-sentence with full document presentation would be most effective (I2, I5, I7).

5.4.2 Orientation

Most of the CAT tools currently available to professional translators display source and target sentences side-by-side (left–right). Green et al. (2014a) conjecture that this 'spreadsheet design may not be optimal for reading', proposing a top–bottom arrangement instead. Our results show that top–bottom orientation can indeed be helpful, but not for reading: it significantly accelerates text reproduction (Copy), but provides no advantage over left–right orientation in tasks that involve no (Scan) or little writing (Revise). Conversely, left–right orientation enables faster revision, significantly so in the Named Entity subtask (Table 5.4).

Top–bottom orientation is surprisingly popular among subjects. S9 comments that 'I first thought it might be weird to work like this without having a continuous source text to look at (because the source text is interrupted by the target text), but it worked like a breeze', and S12 notes that in contrast to what they are used to (left–right), 'the orientation is a bit different, but it doesn't bother me'. S19 remarks that 'I have never arranged my programs this way and I might have to'. While only one subject states they prefer the sentence-level UI with top–bottom orientation (ST) before the experiment, 13 out of 20 subjects prefer it thereafter. For reading-intensive tasks, however, it is preferred less often that for the writing-intensive Copy task (Figure 5.4).

As such, our results motivate the use of top–bottom orientation in UIs to support writing. Revision, on the other hand, is faster with left–right orientation (Table 5.4).

5.4.3 Limitations

As motivated in Section 5.1.3, our experimental design maximises internal validity: subjects perform tasks reflecting specific activities that can be relevant when working with CAT tools, but these tasks do not mirror real-life translation in which many activities are interleaved (Ruiz et al., 2008; Dragsted, 2010). Since some of the UIs we explore are not available in widely used CAT tools, conducting an experiment under realistic working conditions has not been an option for our investigation. Our experimental design allows more fine-grained conclusions instead: for example, we can empirically show that the suitability of sentence segmentation is task-dependent, suggesting that the segment-by-segment presentation in CAT tools should not be replaced, but complemented with an unsegmented view of the text. Nevertheless, our findings are pending confirmation in real-life settings, for which an incorporation of our UIs into fully functional CAT tools is required.

Post-experiment feedback on our prototypical UIs may inform this transition. In particular, the scrolling behaviour in document-level UIs turns out to be more important than we anticipated. In our experimental UIs, the source and target text panes have separate scroll bars, and scrolling in one of them does not automatically invoke scrolling in the other. The reason is that since source and target texts may differ in length, distance-based synchronisation – i.e., if the user scrolls down two lines in the source text, also scroll down two lines in the target text – may result in incorrect alignment. Post-experiment feedback suggests that 'linked scrolling of the two panes …would greatly improve productivity' (S18), particularly with the document-level UI that uses a top–bottom arrangement of source and target documents, which S14 called 'a scrolling and matching nightmare'. Surprisingly, the fastest and most accurate results in one of the revision subtasks (Anaphor) are achieved with document-level UIs despite this shortcoming (Table 5.4, DL and DT), and 4 out of 20 subjects state that they find one of these UIs most suitable for the experimental tasks overall (Figure 5.4e).

Lastly, our full-day experiment concentrates on a single language pair (English to German) and domain (news), and involves 22 professional translators (20 plus 2 for a pilot run). Many studies use students or crowd workers instead (e.g., Bowker, 2005; Karimova et al., 2018), and involve a smaller number of subjects and experimental items (e.g., Macklovitch, 2006; Coppers et al., 2018). Nevertheless, we acknowledge that involving further languages, domains, and more subjects would strengthen our results.

With respect to accuracy in particular, the ability of a subject to correct a controlled manipulation, such as a wrong anaphoric relation (Table 5.2a), can be assessed unambiguously (the subject is either able or unable to correct the mistake because there is exactly one valid solution), whereas the overall quality of a translation or revision is difficult to define (House, 2013) and measure (Läubli and Green, 2019), incurs higher expenses, and may ultimately be less insightful than a targeted evaluation. The disadvantage of our experimental design is that it is unclear whether our findings will generalise to real-life translation with non-prototypical (i.e., commercial) CAT tools. Since such CAT tools are not

(yet) available, this cannot be the goal of our investigation; instead, we can empirically show that, for example, sentence segmentation with a top–bottom arrangement of source and target sentences enables significantly faster writing when we factor out source comprehension and target production problems that will occur in any UI (see the description of our COPY task in Section 5.2.1).

5.4.4 Future Work

In future work, our experimental UIs should be integrated into richer prototypes or, ideally, a fully fledged CAT tool. Testing the impact of text presentation on translation under realistic working conditions will require many features that are not available in our experimental prototypes, starting with real-time integration of translation suggestions from TMs and/or MT. We have investigated four UIs in three experimental tasks, and our results motivate two avenues for further research in particular: the replacement of left–right with top–bottom orientation in sentence-level interfaces, and the use of document-level interfaces for revision.

We consider the latter to be important since the quality of machine-generated translation suggestions is improving steadily (e.g. Junczys-Dowmunt, 2019), which may reduce the amount of writing needed to produce publication-quality translations and in turn increase the need for UIs that are optimised for revision.

Our study also sheds light on how experience with MT affects accuracy in professional translators. Regular users of MT detect significantly more errors within sentences than occasional or non-users (Section 5.3.2), but are the least accurate in revising incoherently translated named entities across sentences (Section 5.3.3). Future work will have to investigate whether the strong focus on single sentences in MT system outputs – and/or in the UI layout of CAT tools – has a priming effect on professional translators.

5.5 Summary

In a controlled experiment with 20 professional translators, we test the impact of changes in text presentation on speed and accuracy in three text processing tasks. We find that:

- Sentence-by-sentence presentation enables faster text reproduction (COPY) and within-sentence error identification (SCAN) compared to unsegmented text; it does not enable faster revision (REVISE).
- Presentation of documents (unsegmented text) leads to the highest accuracy in revision for anaphoric relations between sentences (REVISE, Anaphor).
- Top–bottom orientation of source and target sentences enables faster text reproduction (COPY) than left–right orientation, and is preferred by the majority of subjects in all experimental tasks.

- Left–right orientation enables faster revision for lexical cohesion (Revise, Named Entity).

Our results suggest that the impact of text presentation has been overlooked in the conception of translation technology. Widely used CAT tools implement sentence-by-sentence presentation with left–right orientation for both translation and revision, but our measurements and feedback from subjects imply that source and target sentences should be presented in a top–bottom arrangement, and that CAT tools should offer a side-by-side view of unsegmented text for revision. Most commercial systems do not support these UI layouts, and should be revisited as, at least within the scope of our controlled experiment, they have have a significant impact on translator performance.

Chapter 6

Incorporation of Prior Knowledge into MT Output

While advances in neural modelling have improved MT quality (Section 2.2), machine-translated texts still contain errors that need to be corrected before publication (Chapter 3, Table 3.5). In professional workflows, some of these errors can be anticipated. Consider the case where a translator is asked to translate the English software manual for a new version of an email client into German. The manufacturer may provide the translated manual of the software's previous version, and ask the translator to use the same terminology in the new version to avoid confusion among German-speaking users. The manufacturer may also ask the translator to reuse as much of the translations in the previous manual as possible to save on time and ensure consistency. If the manufacturer provides the translated segments and terminology in a TM and TB alongside the English version of the software manual to be translated, the translator will want to use a CAT tool to leverage these resources while translating the manual into German. Sentences that are the same in the previous and new version of the manual can be retrieved from the TM. For sentences that only changed slightly in the new version, the translator will use the translation suggestions (fuzzy matches) which the CAT tool offers from the TM, but for sentences with major changes or no equivalence in the previous version, using translation suggestions from MT will be more efficient (Sánchez-Gijón et al., 2019). However, while MT suggestions can be integrated into many CAT tools such that they are displayed among other translation suggestions (such as fuzzy matches), they are generated outside of the CAT tool and cannot benefit from the TM and TB or the translator's knowledge. Although the TB may specify that 'spam' be translated as 'Werbemails', a regular MT system will not be able to use this information and likely translate the term as 'Spam', 'Spam-Mails', or 'ungewünschte E-Mails',[1] and the translator will need to correct these instances whenever a source sentence contains 'spam'. Similarly, the TM may contain sentences from the old manual that overlap in part with sentences in the new manual, and MT could be used to

[1] 'Werbemails' is a valid, but not the most frequent translation of 'spam' in German, so it is not likely that a regular German–English MT system would use this translation.

translate the non-overlapping parts rather than the entire segment. Finally, the translator may want to use MT while translating a sentence, e.g., in the form of a completion for a partially translated sentence, where the MT system should use the user-generated parts as-is and only translate the parts that the user has not translated yet.

Most MT systems currently available to translators do not allow for such interactions, and the lack of possibilities to influence their output 'leads to a level of mistrust and sometimes also to rejection of the technology' among professionals (O'Brien, 2012). Just as the name suggests, PE does not allow users to incorporate prior knowledge into MT systems; the systems translate entire source texts before translators can (or have to) intervene. This is contrary to user requirements: Moorkens and O'Brien (2017) ask professional translators if they would find 'context dependent completions' to their input from MT systems useful, which 93 % (215 out of 231 participants) answer in the affirmative, provided that the feature can be turned on and off as needed.

In this chapter, we focus on methods to incorporate prior knowledge in the form of partial translations, hereafter referred to as constraints, into the output of MT systems. Note that we do not discuss methods that adapt MT models based on user input over time (e.g., Peris et al., 2017; Wuebker et al., 2018). In use cases where constraints are known upfront and/or change frequently, it is useful to avoid a learning period and make sure that these constraints will be reflected in system outputs right away. We describe existing methods and introduce a new method to achieve this behaviour in Sections 6.2 and 6.3, respectively, and discuss the strengths, weaknesses, and suitability for different use cases in Section 6.4.

6.1 Use Cases

We describe three use cases in the context of professional translation in which constraining the output of MT systems can be useful. We give a brief description, provide an example, and list opportunities and challenges for each use case.

6.1.1 User Input

In typical PE workflows, the translator gets a full translation suggestion from an MT system for each sentence to be translated. As introduced in Section 2.3.1, the idea of IMT is to continuously provide suggestions as the translator is typing. To make sure that what the translator has already translated themselves is not overwritten by the MT system, their input is treated as a constraint in generating system outputs (suggestions).

Suggestions can take various forms. In the sentence completion paradigm (Esteban et al., 2004; Barrachina et al., 2009), the translator's input is always treated as the beginning of a sentence, and the MT system generates a suggestion that starts with the given input, referred to as the prefix (Table 6.1a).

Source:	2. Choose an attachment, then click Choose File.
User Input:	2. Wählen Sie einen Anhang
Output:	2. Wählen Sie einen Anhang und klicken Sie dann auf Datei auswählen.

(a) User input (sentence completion)

Source:	2. Choose an attachment, then click Choose File.
Output_1:	2. Wählen Sie eine Anlage und klicken Sie dann auf Datei auswählen.
User Input:	2. Wählen Sie eine Anhang und klicken Sie dann auf Datei auswählen.
Output_2:	2. Wählen Sie einen Anhang und klicken Sie dann auf Datei auswählen.

(b) User input (pick-revise)

Source:	2. Choose an attachment, then click Choose File.
User Input:	2. Wählen Sie *, klicken Sie dann auf "Datei auswählen".
Output:	2. Wählen Sie einen Anhang, klicken Sie dann auf "Datei auswählen".

(c) User input (gap filling)

Source:	2. Choose an attachment, then click Choose File.
Termbase:	Attach button → Taste "Anhang", attachment → Anhang, . . ., Choose File → Datei wählen, . . .
Output:	Wählen Sie einen Anhang und klicken Sie dann auf "Datei wählen".

(d) Termbase

Source:	2. Choose an attachment, then click Choose File.
Fuzzy Match:	4. Choose a picture, then click Choose File. 4. Wählen Sie ein Bild aus, und klicken Sie dann auf "Datei wählen".
Output:	2. Wählen Sie einen Anhang aus, und klicken Sie dann auf "Datei wählen".

(e) Fuzzy match

Table 6.1: Use cases for inclusion of prior knowledge (underlined) into MT output.

Whereas prefix decoding always involves a single constraint with a known position, other use cases are more challenging because they may involve multiple constraints whose position in the output is not known upfront. In the pick-revise paradigm (Cheng et al., 2016), the translator starts with unconstrained MT output. They then identify (i.e., pick) and revise the most critical error, which is henceforth considered a constraint, and a new suggestion is generated; this process repeats until the translator sees fit (Table 6.1b).

We also introduce a third interaction paradigm, gap filling, in this chapter (Table 6.1c). Whereas the pick-revise paradigm starts with unconstrained MT that may prime the translator (Green et al., 2013; Moorkens et al., 2018), the idea here is to let a translator translate from scratch, and leave a placeholder (denoted as * in this chapter) wherever they want MT to suggest a completion. The translator's motivation for leaving a gap could be that they do not know how to translate a specific word or phrase but have a plan for the rest of the sentence, and want to type it out to avoid forgetting about it while looking up the aforementioned word or phrase; or that they consider some part of a sentence trivial enough for MT and only want to 'dictate' the parts they consider crucial or not suitable for MT, such as idioms or domain-specific terms. The MT system will treat all user inputs as an ordered sequence of constraints, and generate a translation that fills the gaps between these constraints.

Incorporating user input into MT output will increase the perception of control over the MT system, the lack of which is often described as a factor for non-adoption of MT among professional translators (O'Brien, 2012; Cadwell et al., 2016). IMT paradigms that are not based on an initial MT suggestion, such as sentence completion and gap filling, may also lead to more diverse translations since translators are, depending on the exact implementation of the protocol, not primed by MT (Green et al., 2013; Moorkens et al., 2018). However, decoding speed is crucial because translation suggestions must possibly be updated after every keystroke, and provided within a few hundred milliseconds to be appreciated and used by translators (Green et al., 2014a), especially if they are touch typists (Alabau et al., 2015).

6.1.2 Terminology

TBs, collections of terms associated with translations and metadata such as definitions or usage examples, are commonplace in professional translation (Section 2.1.1). When a translator is tasked to translate specific words or phrases (terms) in a particular way, these specific translations can be stored in a TB and included in the work package that the translator imports into their CAT tool. The CAT tool will typically identify terms in the source sentence that is being translated, and display their translation found in the TB – if any – in a dedicated UI pane (see the Term Recognition window in Figure 2.12). CAT tools do not (yet) propagate terms to MT systems, and the translator has to verify manually that each system output uses correct terminology. If the MT system allows the inclusion of partial translations, the translations of terms identified in the source sentence (the input) could be used as constraints on the output (Table 6.1d).

Compliance with domain-specific terminology is an open problem in state-of-the-art MT (e.g., Haque et al., 2019; Yamada, 2019), and mistranslated terms in MT have a negative impact on the quality of post-edited texts (Vardaro et al., 2019).[2] Treating term translations as constraints could alleviate this problem, and has the potential of reducing the need for model adaptation: if limited resources prohibit the training of domain-specific models, a generic model could be combined with a domain-specific TB instead. A major challenge, on the other hand, is that terms are stored in TBs in their base form and may need inflection when embedded in a sentence. As such, terms will have to be treated as fuzzy constraints, which, in contrast to exact constraints such as user input (see above), can take a modified form in the MT output.

6.1.3 Fuzzy Matches

When translators work with TMs, CAT tools check if there are exact or similar (fuzzy) matches for a sentence that is being translated in the TM (Section 2.1.1). If there is an exact match, MT is typically not needed, but if there is a fuzzy match, the translator has to decide whether post-editing the fuzzy match or MT is more efficient. Fuzzy matches contain partial translations that could be used to constrain the output of an MT system, making sure that the latter only translates the parts of the sentence that are not covered by the fuzzy match (Table 6.1e). This is similar to the gap filling use case with user input (Table 6.1c), the difference being that the ordered sequence of constraints is extracted from a fuzzy match rather than generated by the user.

As with terminology, the partial translations leveraged from a domain-specific TM can be expected to be more appropriate than what a general-domain MT system would generate. The combination of fuzzy matches with MT thus has the potential to enable domain-specific MT without the need for model adaptation, and to reduce the amount of editing needed on part of the translator. Challenges include the extraction of relevant parts from the target side of fuzzy matches, and deciding whether the MT system should treat them as exact or fuzzy constraints, i.e., whether the extracted parts should appear in the MT output as-is or with modifications (if needed).

6.2 Existing Methods

We now describe existing methods to influence the output of NMT models. While we also summarise the published evaluation methods and results for each method, the discussion of their suitability for the use cases described in the previous section is deferred to Section 6.4. Note that our review is not limited to methods that guarantee the inclusion

[2] In a study assessing the quality of post-edited English to German translations at the European Commission's Directorate-General for Translation (DGT), the authors find a positive correlation between the number of terminology errors and mistranslations – which they treat as a single category because their distinction proves to be 'rather difficult' – in raw MT and post-edited MT: 'For every mistranslation or terminology error in NMT, there were 0.03 errors in NMTPE' (Vardaro et al., 2019).

Constraint	Constraint Symbols (BPE)	Constraint Symbols (SentencePiece)
Haus	Haus	_Haus
Werbemails	Werbem@@ ails	_Werbe mail s
vorausdenkende Führungsperson	voraus@@ denkende Führungsper@@ son	_voraus denken de _Führung s person

Table 6.2: Examples of output constraints (German) as segmented by a BPE (Sennrich et al., 2016b) and a SentencePiece (Kudo and Richardson, 2018) model.

of all constraints specified for an output,[3] to which we refer as enforcement methods; we also describe methods that aim at, but do not guarantee, the insertion of constraints, and refer to them as provocation methods. As described in the previous section, the insertion of constraints into the MT output can either be exact or fuzzy in the sense that the constraints will appear exactly as-is or with modifications (such as morphological inflection) in the output, respectively.

Throughout the remainder of this chapter, we use the following notation for all method descriptions. A constraint c is a sequence of N constraint symbols $y_p, \ldots, y_{p+N-1}$ to be included in an output $y = y_1, \ldots, y_T$ of length T. Its position p in the output y is unknown prior to decoding. The constraint symbols correspond to the model's vocabulary and will typically be subwords (Section 2.2.2), as shown in Table 6.2. We denote the constraints to be included in an output y as a set $C = \{c_1, \ldots, c_N\}$ or an array $C = [c_1, \ldots, c_N]$ depending on whether their order in y is unknown or known before decoding, respectively. For example, {Systemvoraussetzungen, Rechner} means that y should contain the words 'Systemvoraussetzungen' and 'Rechner' at any position, in any order. Some methods condition the inclusion and placement of constraints on the presence of certain words in the input $x = x_1, \ldots, x_S$. We use the $\rightarrow$ symbol to denote such dependencies: {system requirements $\rightarrow$ Systemvoraussetzungen} means that the constraint 'Systemvoraussetzungen' on the output y is associated with 'system requirements' in the input x.

6.2.1 Masking

Crego et al. (2016) mask constraints with a placeholder symbol in the training data. A regular encoder-decoder model – the authors use a bi-RNN model with attention (Section 2.2.2) – is trained to translate these placeholders as-is, and at test time, the placeholders generated in the output are replaced with the specified constraints in a post-processing step. An example is shown in Table 6.3.

[3] In that sense, the term 'constraint' may be misleading, but we consider it more concise than 'partial translation'.

Source:	2. Choose an attachment, then click Choose File.
Constraint:	{attachment → Anhang, Choose File → Datei auswählen}
Input:	2. Choose an ENTITY1, then click ENTITY2.
Output:	2. Wählen Sie ein ENTITY1 und klicken Sie dann auf ENTITY2.
Post-proc.:	2. Wählen Sie ein Anhang und klicken Sie dann auf Datei auswählen.

Table 6.3: Masking (Crego et al., 2016).

The authors do not evaluate their approach empirically,[4] but a theoretical weakness is that since regular encoder-decoder models do not guarantee that each input word is covered in the output (Tu et al., 2016), placeholders can be omitted during decoding, such that the corresponding constraints cannot be inserted in the post-processing step. Crego et al. guard against this case in that they 'make sure that an entity placeholder is translated by itself in the target sentence' during beam search, but give no further details. Since the model is unaware of what a placeholder represents during decoding – ENTITY1 could become any word or sequence of words in the post-processing step – the symbols generated around placeholders may not be compatible with the constraints they will be replaced with – masking is an exact method by nature, meaning that constraints are placed in the output as-is, and may need further adjustment (such as morphological inflection) in the post-processing step. An advantage, on the other hand, is that the method does not add overhead in decoding speed, regardless of the number of constraints.

6.2.2 Constrained Autoregressive Decoding

Recall from Section 2.2.2 that given an input sequence x, the probability of an output sequence y under a parametrised model is defined as

$$P(y|x) = \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x). \tag{6.1}$$

The number of possible output sequences, $|V|^L$, is determined by the number of output symbols $|V|$, the model's vocabulary size, and by the maximum length of a sequence L. While scoring all $|V|^L$ possible output sequences is intractable, greedy selection of the most probable output symbol $\hat{y}$ at each timestep, i.e.,

[4] Masking is one among many features introduced in Crego et al.'s (2016) article which is not evaluated independently. A qualitative human evaluation of 50 system outputs (sentences) finds 10 issues (7 major, 3 minor) with named entities, but it is unclear whether all of them were masked during translation.

$$\hat{y}_t = \arg\max_{y \in V} p(y|y_1, \ldots, y_{t-1}, x), \tag{6.2}$$

prevents us from selecting output symbols that are sub-optimal at the current timestep, but will lead to a better score overall. As a compromise between exhaustive search and greedy selection, beam search explores k instead of all (exhaustive) or one (greedy) hypotheses at each timestep t: the probability distribution over all possible continuations (each $y \in V$) is obtained from the model's output distribution for each hypothesis y^k formed in the previous timestep $t-1$, and the k highest scoring hypotheses are kept in the current timestep t, i.e.,

$$\hat{y}_t = \text{k-arg}\max_{y_t^k \in V} \prod_t p(y_t^k|y_1^k, \ldots, y_{t-1}^k, x). \tag{6.3}$$

Hypotheses containing the end-of-sequence symbol are finished and not continued in subsequent timesteps.

The idea of constrained decoding with modified beam search is to enforce the continuation of hypotheses with constraint symbols rather than the most likely symbols according to the model's distribution learned from training data.

Prefix Decoding

If we want to include a single constraint at the beginning of an output, i.e., a prefix, the modification to beam search is straightforward: we chose the constraint symbols in the prefix $c = c_1, \ldots, c_N$ over the model's predictions, one at each timestep, until all symbols are present in the output, and then proceed with regular beam search until the end-of-sentence symbol is generated in all hypotheses:

$$\hat{y}_t = \begin{cases} c_t \text{ if } t \leq N, \\ \text{k-arg}\max_{y_t^k \in V} \prod_t p(y_t^k|y_1^k, \ldots, y_{t-1}^k, x) \text{ otherwise.} \end{cases} \tag{6.4}$$

As described in more detail in Section 2.3.1, Knowles and Koehn (2016) evaluate this method with a bi-RNN model with attention (Section 2.2.2) to generate continuation suggestions for the given beginning of a translation, and achieve better prediction accuracy compared to a phrase-based statistical MT baseline (Ding et al., 2016).

Unlike masking (Section 6.2.1), prefix decoding requires no modification at training time – the method can be used with any model that generates output sequences from left to right (Equation 6.1). It also results in no loss of decoding speed; on the contrary, at timesteps where a constraint symbol is enforced, the model predictions can be shared across beams because the history will be the same in all hypotheses, or not computed at all if the model's score for the prefix constraint is not of interest. However, as the name

suggests, prefix decoding cannot be used with multiple constraints, or a single constraint that may be placed at any position in the output.

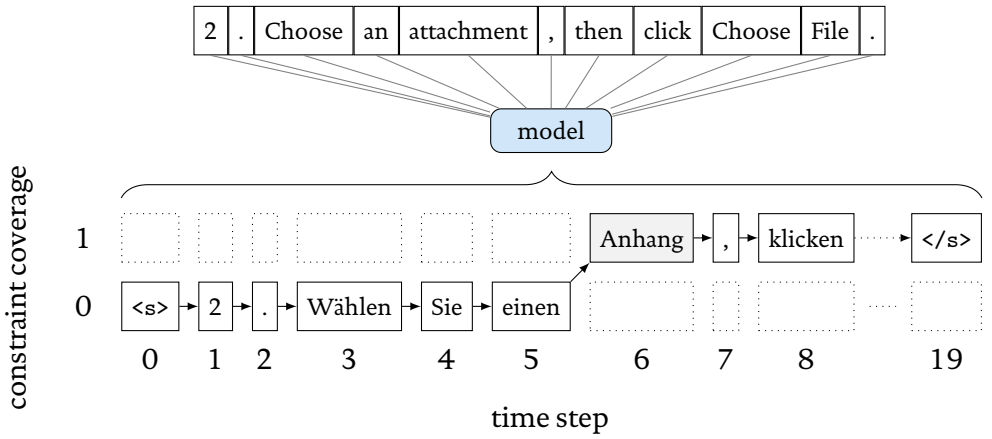
Grid Beam Search

Hokamp and Liu (2017) introduce Grid Beam Search (GBS), a method that allows the inclusions of any number of constraints whose optimal position in the output is unknown prior to decoding. While, graphically speaking, beam search organises hypotheses in a list where each element corresponds to a timestep and holds k hypotheses, GBS adds a second dimension to account for the number of constraint symbols covered in each hypothesis. As such, hypotheses are organised in a grid of $t \times (|C| + 1)$ cells, where $|C|$ is the total number of constraint symbols in the set of constrains C. Each cell holds k hypotheses, and each hypothesis is either closed or open: closed means that the latest symbol added to the hypothesis is part of a constraint c, so it can only be followed by the next constraint symbol in c; open means that the latest symbol added to the hypotheses is either the last symbol of a constraint or not part of a constraint at all, so the next symbol can either be the first symbol of a new constraint or a symbol chosen from the model's distribution.

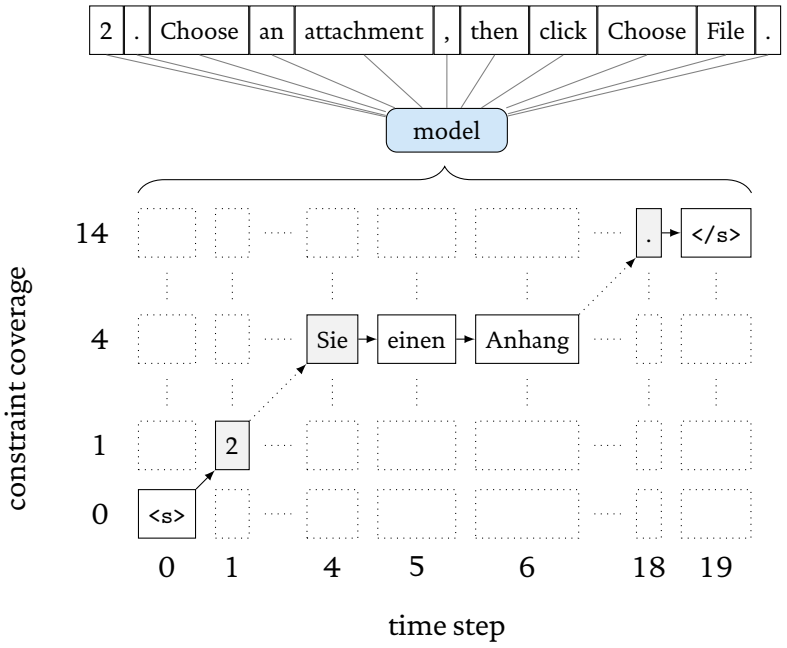
The cells, each holding k hypotheses, are populated by means of dynamic programming: in each cell Grid$[t][j]$, we obtain candidates by (i) generating continuations for the open hypotheses in Grid$[t-1][j]$ from the model's distribution; (ii) adding the first symbol of a new constraint to the open hypotheses in Grid$[t-1][j-1]$; and (iii) adding the next constraint symbol to closed hypotheses in Grid$[t-1][j-1]$. We keep the k highest scoring candidates in each cell. Hypotheses at the top level of the grid cover all constraints, and are added to the set of finished constraints once they generate the end-of-sequence token. The best-scoring hypothesis in this set is the best output that includes all constraints, according to the parametrised model. An illustration is shown in Figure 6.1. Note that we only show the added symbol in the best-scoring hypothesis in each cell, and omit symbols that are not part of the best-scoring output.

GBS guarantees that all constraints are placed in the output. It has has been shown to perform favourably with simulated user input in the pick-revise paradigm (Cheng et al., 2016, Table 6.1b) and in an on-the-fly domain adaptation scenario (Hokamp and Liu, 2017). The authors train an MT model on news data, extract terminology – n-grams with high pointwise mutual information scores (Church and Hanks, 1990) – from a software localisation corpus, and combine the two using GBS to translate held-out test data from the software localisation corpus. The terminology-induced translations achieve higher BLEU scores not only compared to the translations produced by the news model without GBS, but also compared to modifications of these translations where the constraints are inserted at the beginning or at a random position, which demonstrates the ability of GBS to place constraints at suitable positions.

A drawback of GBS is that decoding time increases linearly with the number of constraint symbols. In use cases with many and/or long constraints (Figure 6.1b), GBS



(a) $C = \{$ Anhang $\}$



(b) $C = \{$ 2 . Wählen Sie, , klicken Sie dann auf " Datei auswählen " . $\}$. Note that the comma after 'Sie' separates the two constraint sequences (a prefix and a suffix); the second comma is part of the suffix.

Figure 6.1: Grid Beam Search (Hokamp and Liu, 2017) with few (Figure 6.1a) and many (Figure 6.1b) constraint symbols to be included in the output. The latter increases the search space and thus leads to slower decoding.

explores a much larger search space than in use cases with few and/or short constraints (Figure 6.1a).

Dynamic Beam Allocation

Post and Vilar (2018) note that another disadvantage of GBS is that the effective beam size – the number of hypotheses kept at each timestep – changes between sentences unless they are to incorporate the same number of constraint symbols, which complicates beam search optimisations such as batched decoding. The authors propose an alternative to GBS, Dynamic Beam Allocation (DBA). Hypotheses that cover the same number of constraint symbols are grouped into banks, similar to the rows in a GBS grid. Unlike GBS, however, the number of hypotheses kept in each bank varies: DBA divides a fixed k-sized beam across banks at each timestep, such that the runtime complexity is the same as in regular beam search.

The two fundamental operations in DBA are candidate generation and beam allocation. At each timestep t, the set of candidates is constructed by extending all hypotheses from the previous timestep $t-1$ with the model's unconstrained output distribution, and selecting (i) the k highest-scoring symbols across these hypotheses (Equation 6.3); (ii) the highest scoring symbol for each individual hypothesis (Equation 6.2); and (iii) all unmet constraint symbols from each individual hypothesis. In the beam allocation step, these candidates are grouped into banks according to how many constraint symbols they cover, and the number of candidates kept in each bank is defined as $k/(|C|+1)$, where any remainder is allocated to the bank that covers all $|C|$ constraint symbols. Since this allocation strategy can leave banks underfilled, Post and Vilar use a bank adjustment procedure that is not detailed in their publication.[5] Hypotheses are finished when they generate the end-of-sequence symbol, which is only allowed once they have generated all constrained symbols.

Post and Vilar (2018) empirically show that DBA outperforms GBS in terms of decoding speed. BLEU scores, on the other hand, are slightly lower in DBA overall, although a direct comparison is difficult since k is fixed in DBA and varies among sentences in GBS (see above). DBA requires a large beam[6] and hypothesis pruning for optimal performance (ibid.).

6.2.3 Constrained Non-autoregressive Decoding

Sequence generation with the MT architectures introduced in Section 2.2.2 and employed by the constrained decoding methods described above is autoregressive: outputs symbols are generated from left to right, one at a time, so a symbol at timestep t cannot be

[5] The code is publicly available as part of Sockeye 2 (Hieber et al., 2020).

[6] Dinu et al. (2019) find that DBA at $k = 5$ only places 82 % of the specified constraints in a terminology use case. The authors note that performance can be increased to 99 % by using $k = 20$, but this 'results in a drastic latency cost'.

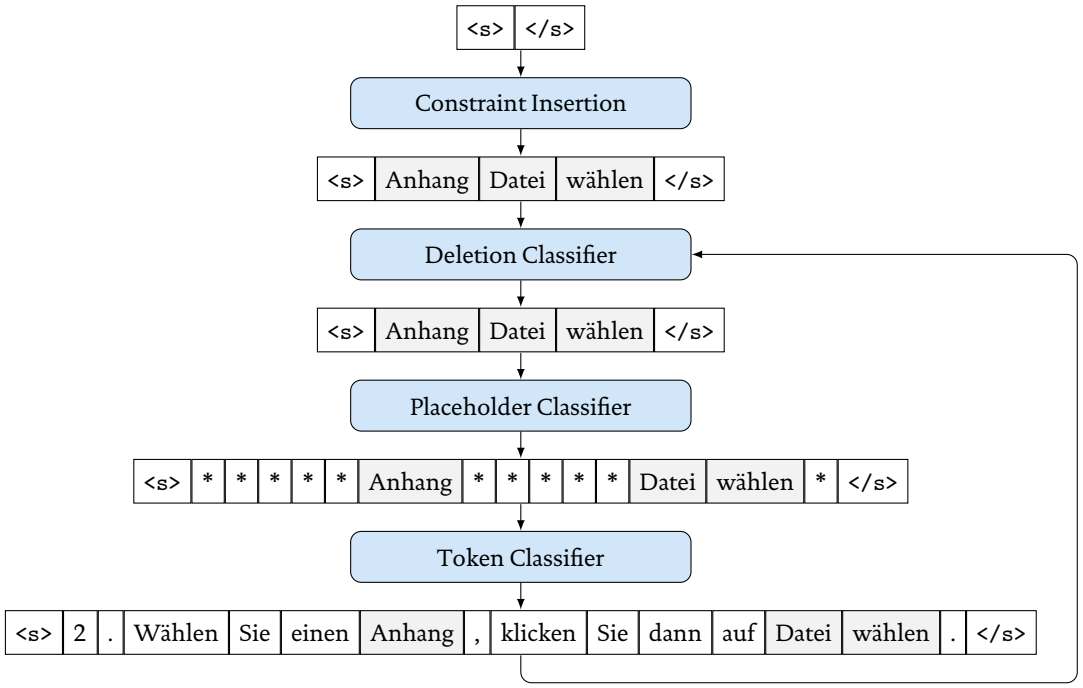


Figure 6.2: Constrained Levenshtein Transformer (Susanto et al., 2020).

generated until the symbol at timestep $t-1$ has been generated (Equation 6.1). The idea of non-autoregressive MT (NAT) is to generate output tokens in parallel (Gu et al., 2018). While Gu et al.'s (2018) approach increases decoding speed by an order of magnitude, it does not achieve the output quality of an autoregressive baseline (ibid.). This is not surprising given that some linguistic features of words in a translation can not be inferred from the source sentence alone, but depend on coordination with other words in that translation. To this end, recent NAT approaches treat output generation as a process where the possibly suboptimal words in an initial translation y^0 are refined iteratively (Lee et al., 2018). Gu et al.'s (2019) Levenshtein Transformer (LT), for example, modifies an output y^m to form a new output y^{m+1} via a sequence of three classification steps: (i) a deletion classifier predicts the symbols that should be deleted, if any; (ii) a placeholder classifier predicts the number of new symbols that should be inserted between any two existing symbols, if any, and inserts placeholders for these new symbols; (iii) a token classifier replaces each placeholder with a symbol from the model's vocabulary. Each prediction to form y^{m+1} is conditioned on y^m and the encoded source sentence e, and decoding proceeds until the output does no longer change (i.e., $y^{m+1} = y^m$) or a maximum number of refinement operations is reached.

Susanto et al. (2020) adapt Gu et al.'s (2019) LT decoder to incorporate output constraints (CLT, Figure 6.2). Whereas y^0 in LT is a sequence that only contains the beginning-of-sequence and end-of-sequence tokens, Susanto et al. insert all constraints into y^0. In the decoding loop, the deletion classifier is not allowed to delete constraint symbols, and the placeholder classifier is not allowed to insert placeholders between any two constraint

symbols. Susanto et al. (2020) evaluate CLT against DBA and the data augmentation method introduced by Dinu et al. (2019, described below) in a terminology use case, where CLT achieves substantially higher BLEU scores.[7] CLT also achieves a term usage rate of 100 %, meaning that all of the specified constraints are actually placed in the output, compared to 76 % (DBA) and 95 % Dinu et al. (2019) in one of the two evaluated settings.

A limitation of Susanto et al.'s (2020) CLT implementation is that constraints are inserted in the order in which they appear in the source sentence, and since the deletion classifiers disallows the deletion of constraint symbols, constraints cannot be reordered in the output. The authors find that the order of constraints (terms) is the same in 97–99 % of the source and target sentences in their English to German test set, but note that the issue of constraint reordering 'may become more apparent in language pairs with more distinct syntactic differences' (ibid.).

6.2.4 Data Augmentation

The constrained decoding methods described in Sections 6.2.2 and 6.2.3 use exact insertion, i.e., constraints appear in the output without any modification. In some use cases, however, we want to allow the modification of constraints. For example, we may want to allow the morphological inflection of terms that are only available in their base form (Section 6.1.2).

In this section, we describe methods that encode constraints together with the source sentence to be translated, and train models that will include these constraints in the output without modifying the decoding process. While these methods do not guarantee that the encoded constraints will appear in the output – they can be characterised as provoking rather than enforcing the specified constraints – they may implicitly learn to apply meaningful modifications to the specified constraints, as outlined below.

Input Features

Factored MT (Koehn and Hoang, 2007) allows the annotation of input words with arbitrary features, such as part-of-speech tags. The motivation is that the additional information may help disambiguate the input words, and ultimately guide the model to produce better translations. The English input word 'building', for example, could be translated into German as 'Haus' or 'bauen', depending on whether it is used as a noun or as a verb in the input, and associating it with a source factor such as 'NOUN' or 'VERB' would resolve this ambiguity. In NMT, source factors can be embedded like input words,[8] using a

[7] However, Susanto et al. (2020) use a deeper model with more parameters than Post and Vilar (2018) and Dinu et al. (2019), which likely attributes to some of the difference in BLEU.

[8] Words and other input features such as part-of-speech tags do not differ from a technical perspective, they just use different vocabularies.

Source:	2. Choose the desired attachments, then click Submit.
TB:	Attach button → Taste "Anhang", attachment → Anhang, ...
Input (App.):	2 . Choose the desired attachments Anhang , then click Submit . 0 0 0 0 0 1 2 0 0 0 0 0
Input (Rep.):	2 . Choose the desired Anhang , then click Submit . 0 0 0 0 0 2 0 0 0 0 0
Output:	2 . Wählen Sie die gewünschten Anhänge , klicken Sie dann auf Absenden .

Table 6.4: Source factors (Dinu et al., 2019).

separate vocabulary and embedding matrix, and concatenated with input words prior to encoding (Sennrich and Haddow, 2016).

Dinu et al. (2019) augment source sentences with output constraints (Table 6.4). Since the model input thus contains words from both the source and the target language, the authors use a source factor to disambiguate the code switching with a feature value of 0 for normal input words, 1 for words of the source language that should be replaced by a constraint, and 2 for words of the target language (i.e., constraint symbols).

At training time, Dinu et al. (2019) construct training examples from a regular parallel corpus and a TB. Terms are matched in the sentences of the parallel corpus using approximate string matching to allow for some morphological variation – such as matching 'attachment' in the TB for a source sentence containing 'attachments' (Table 6.4) – and the translation of each matched term is either appended to (App.) or used instead of (Rep.) the corresponding source words in the source sentence. The output sentence is left unchanged, and the hope is that the MT model will learn to copy the target words in the augmented source and translate all other words as usual. To ensure that the model will also learn to translate inputs without constraints, some of the terminology matches are ignored at random during training.

Dinu et al. (2019) demonstrate the effectiveness of their method in separate experiments with two publicly available TBs: Wiktionary and IATE, the TB used in institutions of the European Union. On test sets with sentences and terms not used during training, the method shows slightly higher BLEU scores than a baseline model trained on the same parallel corpus, but without the term annotations. DBA achieves a higher term usage rate in the Wiktionary test set, whereas Dinu et al.'s (2019) method method achieves a higher term usage rate in the IATE test set; DBA achieves lower BLEU scores in both test sets, indicating that the placement of constraints in the output with DBA may succeed, but at the cost of lower translation quality. However, these test sets are constructed such that the translations in the TB are exact matches in the source and target sentences. Dinu et al. (2019) also conduct an evaluation with target sentences that contain approximate matches of the term translations, where their method achieves significantly higher BLEU scores than DBA both with terms from Wiktionary and IATE.

Source:	2. Choose an attachment, then click Choose File.
Fuzzy Match:	4. Choose a picture, then click Choose File. 4. Wählen Sie ein Bild aus, und klicken Sie dann auf "Datei wählen".
Input:	2 . Choose an attachment , then click Choose File . @@@ 4 . Wählen Sie ein Bild aus , und klicken Sie dann auf " Datei wählen " .
Output:	2. Wählen Sie einen Anhang aus, und klicken Sie dann auf " Datei wählen " .

Table 6.5: Neural Fuzzy Repair (Bulté and Tezcan, 2019).

Due to the use of approximate string matching in training data generation, the model is presented with examples in which an inflected source term and the base form of a target term in the input are paired with an inflected form of that target term in the output, such as attachments/Anhang → Anhänge in the (theorised) example shown in Table 6.4. Dinu et al. (2019) observe that 'in some cases, our models generate morphological variants of terminology translations provided by the term base', but do not investigate this phenomenon empirically.

Input Concatenation

Data augmentation can also be used to leverage partial translations contained in TMs. Bulté and Tezcan (2019) retrieve the fuzzy matches for a source sentence to be translated from a TM, and append the translation of the best or n-best fuzzy match(es) to that source sentence (Table 6.5). Note that the method, termed Neural Fuzzy Repair (NFR), does not involve source factors to annotate words in the input. The words of the sentence in the source language and the words of the fuzzy match in the target language are simply concatenated to form the input sequence using a separator symbol ('@@@' in Table 6.5).

In English to Dutch and English to Hungarian experiments using the TM of the Directorate-General for Translation of the European Commission (Steinberger et al., 2012), Bulté and Tezcan (2019) train regular and FMR-enabled MT models on the TM data, and show that the augmentation of unseen sentences with fuzzy matches from this TM (i.e., the training data) leads to substantial BLEU, TER, and METEOR improvements (Section 2.2.3) over the regular system. However, the authors do not empirically assess how much of the original source sentence and fuzzy match(es) are included in the output without modification.

6.3 Infix Generation

We now introduce Infix Generation (IG), a method to generate a sequence of words (the infix) that is compatible with two given constraints: a prefix and a suffix. As will be further

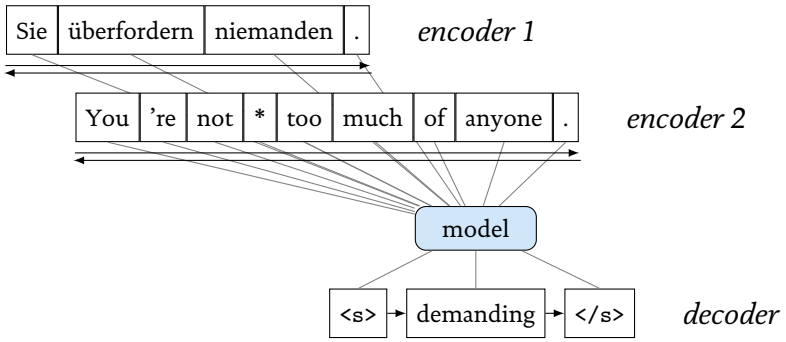


Figure 6.3: Infix Generation (German to English example). The special symbol * is part of the model's vocabulary and marks the infix to be generated by the decoder.

discussed in Section 6.4, we observe that none of the methods described in the previous sections are ideal for the gap filling use case (Table 6.1c), where a user requests a translation suggestion for a word or phrase within a given translation.

6.3.1 Method

As further discussed in Section 6.4.1, one disadvantage in the autoregressive decoding methods described in Section 6.2 is that a decoding step is needed for every constraint symbol to be included in the output sequence. We circumvent this problem by training a model to only generate the symbols that are needed to fill the gap between – and not recreate – two known parts of the output, a prefix p and a suffix s.

We use two separate encoders, one to encode the source sentence to be translated, the other to encode the output constraints p and s. An illustration is shown in Figure 6.3. To obtain training examples, we randomly sample an infix Y from the target side of each sentence pair in a parallel corpus, and train the model to predict the infix given the encoded source, prefix, and suffix. Note that the prefix and suffix can be 'empty' (i.e., only contain the beginning or end of sequence symbol) so as to also allow for suffix generation (i.e., sentence completion), prefix generation, and unconstrained decoding.

More formally, let $x^1 = x_1, \ldots, x_{T_x}$ be a source sequence of length T_x, and $p = p_1, \ldots, p_{T_p}, y = y_1, \ldots, y_{T_y}, s = s_1, \ldots, s_{T_s}$ a constrained target sequence of length T_p (prefix) + T_y (infix) + T_s (suffix). We use x^1, p and s as model inputs. p and s are concatenated using a special symbol φ denoting a placeholder, forming $x^2 = p_1, \ldots, p_{T_p}, \varphi, s_1, \ldots, s_{T_s}$.

We use two separate bi-RNN encoders (Section 2.2.2) for the input sequences x^1 and x^2. Each input sequence of length T_i is encoded as a sequence of annotation vectors $h = h_1, \ldots, h_{T_i}$, where each annotation vector h_t is a concatenation of the forward and backward RNN hidden states at timestep t (Equation 2.21). Our output sequence is the infix y.

For decoding, we use a conditional GRU with attention as introduced by Sennrich

		NP			RANDOM			VERB		
		$\bar{c} = 22.3, \bar{u} = 2.6$			$\bar{c} = 12.2, \bar{u} = 12.3$			$\bar{c} = 23.7, \bar{u} = 1.0$		
		F1	BLEU	Time	F1	BLEU	Time	F1	BLEU	Time
GBS	k=1	49.9	87.0	3.32	52.1	55.1	3.07	31.0	88.9	3.44
IG	k=1	38.6	80.4	**0.16**	40.0	45.4	**0.37**	29.1	86.6	**0.11**
GBS	k=5	47.8	85.2	15.79	**55.1**	**56.6**	15.35	29.0	87.0	15.86
IG	k=5	**53.7**	**87.9**	0.35	48.4	54.7	0.71	**36.3**	**89.3**	0.23

Table 6.6: Experimental results. We compare our method (IG) to Grid Beam Search (GBS) in terms of output quality (F1, BLEU) and decoding time (in seconds per sentence). Metrics are provided for two beam widths ($k = 1, 5$) on three testset variations, differing in the average number of constraint ($\bar{c}$) and to-be-predicted ($\bar{u}$) symbols per output sentence.

et al. (2017). The decoder attends to both encoders, a typical design choice for multi-source models (Zoph and Knight, 2016). At target position j, the context vector c_j is computed by averaging the context vectors c^{x^1} and c^{x^2} from the x^1 and x^2 encoders, respectively:

$$c_j = \frac{c_j^{x^1} + c_j^{x^2}}{2}. \tag{6.5}$$

This combination strategy is referred to as early averaging, and shown to be effective for multilingual translation, by Firat et al. (2016).

6.3.2 Experimental Results

We implement IG into Nematus (Sennrich et al., 2017) and compare it to GBS (Section 6.2.2), leveraging Hokamp and Liu's (2017) original implementation.[9] In the latter, we make two alterations to allow for a fair comparison. First, we enforce constraints to be used in a defined order. This ensures that prefixes and suffixes are always placed at the beginning and end of the generated output sequence, respectively. Second, we add a parameter α to normalise hypothesis scores by length. As in Nematus, the normalised score of a hypothesis is defined as its original score divided by its length$^{\alpha}$. Without

[9] See `https://github.com/chrishokamp/constrained_decoding`. The authors note that their code could be further optimised in terms of speed through parallelisation. The same applies to our implemenation, where the use of a production-oriented MT toolkit would allow for faster decoding. Contrary to our approach, computational cost will increase with the number of constraint symbols in GBS even with optimised code.

Source:	Überzeugen die lächelnden Gesichter?
Constraints:	{ Do, convince? }
Output:	Do **the smiling faces** convince?
Reference:	Do **smiling faces** convince?

Table 6.7: Evaluation example.

some form of length normalisation, shorter hypotheses are favoured over longer ones on average (Wu et al., 2016).

We build German-to-English translation models using all training data from the WMT 2017 news translation task (Bojar et al., 2017).[10] All translations are tokenized, truecased, and segmented into subword units through BPE (Sennrich et al., 2016b). We use word embedding size 512 and hidden layer size 1024, with layer normalisation (Ba et al., 2016) in all feed-forward and recurrent layers except those followed by a softmax. For SGD optimisation, we use adam (Kingma and Ba, 2015) at a learning rate of 10^{-4}. Newstest 2016 is used for early stopping and optimisation of the length normalisation parameter α.

For evaluation, we use three variations of Newstest 2017 (3004 segments). In RANDOM, we extract a single sequence of words of random length at a random position from each English translation, just as with the training data for the multi-source system (see above). In NP and VERB, we extract a random noun phrase or a single verb (excluding auxiliaries), respectively.[11]

We evaluate how well IG and GBS predict the extracted word(s) in each test sentence, given the corresponding source and its partial translation (i.e., the constraints). Translation quality (Table 6.6) is measured by calculating F1 on all predicted words, appearing in bold in the example shown in Table 6.7, and BLEU on full sentences.[12] BLEU scores are high with both methods as the constrained parts overlap in all outputs and references. Nevertheless, we find it to be a useful secondary indicator of translation quality in the task at hand since it rewards matches of multiword sequences. This includes combinations of constrained and predicted words, such as the bigram 'faces convince' in Table 6.7.

IG and GBS achieve similar translation quality (Table 6.6). The latter performs better with longer infixes (RANDOM), whereas our model achieves higher F1 and BLEU scores on shorter, linguistically motivated infixes (NP, VERB). We observe under- and (particularly) over-translation in outputs generated with both methods, a known problem in neural MT (Tu et al., 2016). Another problem is synonymy. As we use a single reference

[10] All training data as well as development and testsets (Newstest 2016 and 2017) are available at `http://statmt.org/wmt17/translation-task.html`

[11] using the PoS tagger and chunker available in spaCy (`https://spacy.io/`)

[12] using the NIST `mteval-v13a.pl` script on detokenized output

for calculating F1 and BLEU on short sequences, correct alternative translations are punished harshly.

We also compare IG and GBS in terms of speed. We run our experiments on an Intel Xeon E5-2650 2.20GHz machine, using a single NVIDIA Titan Xp GPU for each experiment. We measure total wallclock time to process the entire testset, and divide it by the number of sentences per testset to report per-sentence decoding time. IG achieves a speedup of an order of magnitude over GBS, and our results confirm that a higher number of constraint symbols decreases decoding speed in GBS, whereas it increases decoding speed in IG. We discuss the difference in decoding speed between IG and GBS, as well as the other methods described in Section 6.2, in the next section.

6.4 Discussion

6.4.1 Speed

The speed with which constrained outputs are generated is particularly relevant in use cases where users expect an immediate response when interacting with MT (Section 6.1.1), and varies considerably between the methods described in this chapter. In GBS, decoding speed grows linearly with the number of constraint symbols to be placed in the output. In our evaluation (Table 6.6), translating sentences with a high number of constraints results in an average decoding time of more than 15 seconds per sentence at beam size 5. While an optimised implementation of the method would certainly allow for faster decoding, a fundamental problem of the GBS algorithm is that hypotheses that have covered a different number of constraint symbols are not in competition to each other. The same holds for DBA, where hypotheses only compete within banks in which each hypothesis has covered the same number of constraint symbols. The speedup over GBS is owed to the fact that the effective beam size at each time step does not depend on the number of constraint symbols, so the number of hypotheses that cover the same number of constraints explored in the search decreases with the number of constraint symbols. As a result, large beam sizes are needed to guarantee that all constraint symbols are actually placed in the output, which in turn slows down the search (Dinu et al., 2019).

A limitation we find in all of the existing methods that use autoregressive decoding (apart from masking) is that constraint symbols are inserted in separate timesteps. In prefix decoding, for example, we know exactly what the beginning of the output will be, but still need a forward pass for each symbol in the prefix to update the decoder's state; in simplified terms, the model needs to know which words are already covered in the prefix so as to generate a compatible continuation. In IG, we move this information to the encoder, generate the symbols between constraints, and then copy the constraints to the output. While we only experiment with the case where a single sequence is inserted between two constraints – where the method is substantially faster than GBS – in this

chapter, the method could be extended by encoding any number of constraints, and generate one placeholder per constraint rather than all of the constraint symbols to avoid wasting timesteps on known symbols during decoding. In contrast to masking, the hope would be that the generated symbols are morphologically compatible with the placeholders as the output they represent is available to the decoder via the encoded input.

CLT outperforms all of the other methods discussed in this chapter in terms of speed. The roughly 250 constrained sentences per second reported by Susanto et al. (2020) cannot be directly compared to the decoding speed achieved in evaluations of other methods, such as roughly one sentence per second with DBA (Post and Vilar, 2018), not only because Susanto et al. give no information on their hardware setup, but also because now-mainstream optimisation techniques for autoregressive decoding (e.g., Hieber et al., 2020) would allow for faster decoding with GBS and DBA compared to their original evaluation. However, the difference between autoregressive and non-autoregressive decoding methods is of a conceptual nature: in the former, a symbol at timestep t cannot be generated until all preceding symbols have been generated, and the preceding symbols cannot be altered after generation; non-autoregressive decoding generates output symbols in parallel, and the alteration of these symbols is conditioned on all other symbols in the output at any given iteration. As a consequence, there is no significant difference in speed between unconstrained and constrained decoding with non-autoregressive models such as LT/CLT (Susanto et al., 2020), whereas autoregressive methods sacrifice speed to include constraints.

6.4.2 Exact vs. Fuzzy Insertion

Most of the methods discussed in this chapter aim at inserting constraints into the output exactly as-is.[13] In some use cases, this is the expected behaviour: a translator who has produced a translation and requests alternative suggestions for a selected word in that translation may not want the MT system to modify the other words, so these words act as exact constraints (e.g., Table 6.1c). In other use cases, however, the words to be inserted into MT outputs are only available in a normalised form. The terms in a TB, for example, may need inflection when placed into certain outputs (Section 6.1.2).

The degree to which constraints can be inflected in decoding primarily depends on two factors: the decoding method and the model's vocabulary. Constrained decoding methods such as prefix decoding or CLT enforce the presence of constraint symbols in the output, and the constraint symbols themselves will not be modified. The unconstrained symbol appearing immediately after the last symbol of a constraint, in contrast, is sampled from the model's output distribution, and if the vocabulary models word boundaries at the beginning (Kudo and Richardson, 2018) rather than the end of symbols (Sennrich et al., 2016b),[14] this unconstrained symbol could be a suffix to the constrained symbol.

[13] I.e., masking, prefix decoding, GBS, DBA, CLT, and IG.
[14] Examples are shown in Table 6.2.

	a tall building	two tall buildings	she is building a wall
Enforcement			
Exact	ein hohes Haus	zwei hohe Haus [a]	sie baut ein Haus [c]
Fuzzy	ein hohes Haus	zwei hohe Häuser	sie baut eine Hausmauer [c]
Provocation			
Exact	ein hohes Haus	zwei hohe Gebäude [b]	sie baut eine Mauer
Fuzzy	ein hohes Haus	zwei hohe Häuser	sie baut eine Mauer

[a] Wrong grammar [b] Wrong translation [c] Wrong terminology (missing constraint)

Table 6.8: Theorised examples of constrained decoding with {building → Haus}.

For example, the model could insert the symbol 'en' after the constraint symbols '_voraus denken de _Führung s person' to form the plural.

However, the inflection of output constraints will be limited to either suffixation or prefixation in methods that use exact insertion. In cases where inflection cannot be realised through affixation, such as 'Haus' → 'Häuser' in German (Table 6.8), constraints will need to be inflected prior to decoding with a separate mechanism. The method introduced by Dinu et al. (2019), in contrast, implicitly learns morphological inflection during training (Table 6.4), and may thus be the most suitable method for professional translation use cases that involve both MT and TBs. Unfortunately, Dinu et al. (2019) do not show how often the inflection of constraints succeeds in their experiments, so future work will need to explore whether handling morphological inflection outside or within constrained MT systems leads to better accuracy.

One avenue of future research towards modelling constraint inflection within MT sytems would be to extend CLT with an additional classifier. The implementation of Susanto et al. (2020) prohibits the deletion of constraint symbols and the insertion of new symbols between constraint symbols, and a constraint inflection classifier could predict character-level alterations. Dinu et al. (2019) use Levenshtein distance to match terms for creating training examples, and a starting ground in CLT could be to allow constraint alteration up to a certain Levenshtein distance. While these alterations could be linguistically motivated (Wichmann et al., 2010), the model itself may be able to score valid alterations higher than invalid ones. Since constraints to be enforced are not necessarily unknown, but less likely than other translations in a model, a sequence like 'Where are the supplementary component?' may well receive a lower score than 'Where are the supplementary components?', even if the model would produce 'Where are the attachments?' without the {supplementary component} constraint for a given sentence.

6.4.3 Enforcement vs. Provocation

Susanto et al. (2020) highlight that Dinu et al.'s (2019) method does not guarantee that all specified constraints will be placed in the output, whereas CLT achieves a term usage rate of 100 % in their experiments. While the failure to include a specified constraint will be unsatisfactory in IMT scenarios, we note that the ability of a model to ignore all or some of the specified constraints may be beneficial in use cases where large databases of existing translations are leveraged, such as TBs and TMs. We briefly discuss two examples in this regard: ambiguous and unsuitable constraints.

Consider the examples in Table 6.8. A TB may contain ambiguous source terms without further annotations, such as 'building'. While the intent may be to ensure that 'building' as a noun is translated into German as 'Haus', enforcing 'Haus' in translations of source sentences in which 'building' is used as a verb will likely be unintended. In contrast to methods like GBS and CLT, the data augmentation methods described in Section 6.2.4 provoke rather than enforce the presence of constraints in the output, so constraints that are unlikely in a given output (according to the trained model) may not be included. In addition to the example outlined above, this may also be useful when automatic selection of constraints from TMs or TBs results in false positives, e.g., because approximate string matching (as used in Dinu et al., 2019) matches the term 'poodle' in a source segment that contains 'noodle'. However, the negative side effects of not guaranteeing the inclusion of specified output constraints may well outweigh these advantages in real-world use cases, and should be examined in future studies.

6.4.4 Real-time domain adaptation

Ultimately, the choice of a suitable method for incorporating prior knowledge into MT outputs will not only depend on the use case (Section 6.1), but also on whether the goal is maximum control or optimal quality. Research on translator requirements has shown that control is an important factor for (non-)adoption of translation technology among professional translators (O'Brien, 2012; Cadwell et al., 2016, 2018). In particular, translators have more trust in suggestions from TMs than suggestions from MT as 'We have greater control over what comes from our TMs because we are the ones feeding them, so I would trust a hit from the TM', as one of the translators surveyed by Cadwell et al. (2018) puts it. In this light, MT methods that incorporate as much of a fuzzy match as possible without modification and, with proper visualisation, only machine-translate the remainder, or insert a user-defined term exactly as-is even if an inflected version would lead to a higher score according to the model – such as IG or CLT – may lead to better acceptance among professional translators. Provocation methods like FMR that may or may not include a given constraint, in contrast, would likely reinforce the perception of MT as an uncontrollable 'black box' (O'Brien, 2012).

If control is not the first priority, however, these data augmentation methods and FMR in particular open up an opportunity for real-time domain adaptation, where the goal is to

improve a system's ability to produce better translations for texts in a particular domain, regardless of whether the 'parts' of these translations stem from a single or many different existing translations. Bulté and Tezcan (2019) encode multiple fuzzy matches for source sentences to be translated, and achieve remarkable BLEU improvements for unseen sentences that are augmented with fuzzy matches – even if the baseline system has seen all of these fuzzy matches during training. Xu et al. (2020) compare FMR to fine-tuning (Luong and Manning, 2015): they train a regular and an FMR-enabled Transformer model (Section 2.2.2) on the concatenation of parallel corpora from seven different domains, as well as fine-tuned regular systems for each of these domains. At test time, unseen sentences from each domain are either translated with the combined model and domain-specific fuzzy matches (FMR) or with the fine-tuned models. Remarkably, FMR achieves comparable or higher BLEU scores in all seven domains. This opens up an interesting perspective for MT users, such as language services providers, who may have several TMs for different domains or clients, but do not have the means to train domain-specific MT models for each individual TM. Moreover, if the MT system is directly integrated with a TM, segments added to that TM are also immediately available to the MT system, effectively eliminating the need for retraining (hence 'real-time').

6.5 Summary

From enforcing specific translations for single words to real-time domain adaptation, we show that methods of incorporating existing translations into MT outputs are suitable for a wide range of use cases in professional translation. As summarised in Table 6.9, these methods differ in several aspects. Exact methods place partial translations into outputs exactly as-is, while fuzzy methods allow modification such as morphological inflection, which is particularly relevant in terminology use cases. However, the fuzzy methods discussed in this chapter do not guarantee that all specified constraints will necessarily be included in the output, which is an important requirement in IMT use cases such as generating sentence completions or translation alternatives. For these use cases in particular, the speed with which translations can be generated is crucial, and our review shows that existing approaches differ greatly in that regard. We propose a new method (IG) that achieves a significant speedup over a constrained decoding baseline (GBS) at comparable quality. More importantly, however, IG motivates – and demonstrates the feasibility of – encoding constraints that should be placed in the output without modification. Autoregressive constrained decoding methods require a full forward pass in the decoder for each constraint symbol, even if the information obtained in this forward pass – the probability distribution over all symbols in the model's vocabulary at the current timestep – is effectively discarded.

The Constrained Levenshtein Transformer (CLT) as implemented and evaluated by Susanto et al. (2020) also overcomes this limitation, and, as it generates output symbols in parallel rather than in individual timesteps (i.e., non-autoregressive decoding), achieves the highest speed of all methods discussed in this chapter. CLT also outperforms

Method	Reference	Insertion	Presence	Training	$\|C\| \rightarrow$ Speed
Masking	Crego et al., 2016	Exact	Enforcement	Yes	Not relevant
Prefix Decoding	Knowles and Koehn, 2016	Exact	Enforcement	No	Not relevant
Grid Beam Search (GBS)	Hokamp and Liu, 2017	Exact	Enforcement	No	Decrease
Dynamic Beam Allocation (DBA)	Post and Vilar, 2018	Exact	Enforcement [(a)]	No	Not relevant [(a)]
Constrained Levenshtein Transformer (CLT)	Susanto et al., 2020	Exact	Enforcement	No	Not relevant
Input Features	Dinu et al., 2019	Fuzzy	Provocation	Yes	Not relevant
Neural Fuzzy Repair (NFR)	Bulté and Tezcan, 2019	Fuzzy	Provocation	Yes	Not relevant
Infix Generation (IG)	Section 6.3	Exact	Enforcement	Yes	Increase

[(a)] DBA requires large beam size to guarantee presence of constraints (Dinu et al., 2019), which decreases speed regardless of $|C|$.

Table 6.9: Summary of methods discussed in this chapter. $|C|$ is the total number of constraint symbols to be placed in an output.

other methods (Post and Vilar, 2018; Dinu et al., 2019) in terms of quality and constraint coverage, but is currently limited in that constraints cannot be reordered and inflected. As discussed in Section 6.4.3, it may also be necessary to relax the strict non-removal policy for constraint symbols in the deletion classifier in use cases where unsuitable constraints, such as ambiguous terms from a TB, cannot be ruled out. As sketched out in Section 6.4.2, future research could explore the possibility of inflecting constraints in particular, which we believe will be possible within the CLT framework without the need for external pre-processing.

Chapter 7

Conclusion

This thesis set out to gain a better understanding of three key problems with MT in the context of professional translation (Chapter 1). The first key problem is quality, which, as we showed in Chapter 3, is overestimated by researchers and developers (Hassan et al., 2018): state-of-the-art MT does not reach parity with professional human translation. We found that one reason for the difference in how MT researchers and (professional) users perceive quality is that the former have too strong a focus on sentences – regardless of their context within a document – where MT can indeed achieve the same quality as professionals, at least in Chinese to English news translation. However, we showed that this does not hold for full documents, where MT contains significantly more incorrect, missing, and misplaced words, many of which cannot be revealed in a sentence-level evaluation. Our empirical results strongly suggest that MT quality should be evaluated at the level of documents instead of sentences.

The strong focus on sentences also plays a role in CAT tools, the software typically used by professional translators to leverage MT. The way in which translations are visualised in CAT tools, the second key problem explored in this thesis, has been criticised by translation researchers and practitioners (Dragsted, 2006; LeBlanc, 2013; O'Brien et al., 2017). We conducted a controlled evaluation of the fundamental design choices regarding text presentation in CAT tools, and found that they have a significant impact on speed and accuracy in professional translators (Chapter 5). Efforts to increase productivity in translation workflows that involve MT are typically focused on quality (e.g., Green et al., 2014b; Bentivogli et al., 2016), but our investigation shows that the way in which MT is presented – i.e., the design and configuration of CAT tool UIs – should not be left out of the equation.

Third, we investigated methods with which information known before machine-translating a text – such as terminology, fuzzy matches from TMs, or in-situ input from translators – can be used to influence MT output. Correcting the same mistakes over and over again in a translation job frustrates professionals (Macklovitch, 2006), and can be mitigated by enabling MT models to enforce or provoke the placement of user-defined constraints in the output. In existing approaches to autoregressive constrained decod-

ing, a high number of constraints either leads to a decrease in speed or quality, and we introduced an alternative method based on the idea of encoding and thus not having to generate the part(s) of the output that are known upfront (Chapter 6). We evaluated our approach against a constrained decoding baseline (Hokamp and Liu, 2017), and achieved a tenfold speedup at comparable quality.

In the remainder of this chapter, we first give a more detailed summary of the empirical findings presented in this thesis (Section 7.1). We then discuss their practical implications (Section 7.2), and offer methodological suggestions for future research (Section 7.3). Finally, we address the limitations of our work, and outline directions for future work, in Section 7.4.

7.1 Empirical Findings

7.1.1 Quality

Hassan et al. (2018) presented a Chinese to English MT system geared to the news domain. For evaluation, they used this system to translate a number of Chinese news articles into English, and ordered translations for the same articles from professional translators (HT). The authors showed randomly selected sentences from these articles paired with either their machine or professional translation to bilingual crowd workers, and asked them to rate translation quality on a scale from 0 (worst) to 100 (best). We reproduced Hassan et al.'s (2018) evaluation using the same translations,[1] but with a different evaluation protocol: we hired professional translators instead of crowd workers, showed them either random sentences or full news articles (documents) as translated by MT and HT side by side, and asked them to choose the better of the two, with ties allowed.

In accordance with Hassan et al. (2018), we found no statistically significant difference between preference for MT and HT in ratings for isolated sentences. For full documents, in contrast, we found a significant preference for HT. We also collected judgements in a second experimental condition where translators were shown translations without source texts, and found a significant preference for HT over MT on both the level of isolated sentences and full documents. Our results refute Hassan et al.'s (2018) finding of human–machine parity in Chinese to English news translation.

7.1.2 Presentation

Text presentation in widely used CAT tools is guided by two fundamental design choices: the sentences in a document are visually separated (sentence segmentation), and the source and target texts of each sentence are displayed side-by-side (left–right orientation). We measured the impact of these design choices on text processing speed

[1] We excluded articles that were not originally written in Chinese, i.e., translationese.

and accuracy in professional translators, using four experimental UIs that present either segmented text (sentences) or unsegmented text (full documents) in a left–right or top–bottom arrangement.

In our controlled experiment, sentence segmentation enabled significantly faster text reproduction and within-sentence error identification at no loss in accuracy. For revision, on the other hand, sentence segmentation provided no advantage in speed over a display of full documents, which led to the highest accuracy in revision for anaphoric relations between sentences, although the difference is not statistically significant ($p = .06$). Top–bottom orientation led to significantly faster text reproduction, whereas left–right orientation led to significantly faster revision for lexical cohesion. Our results show that text presentation has a significant impact on translator performance.

7.1.3 Adaptability

Constrained decoding methods allow users to define specific words or word sequences (constraints) that must appear in a translation such that the MT system only generates the remaining words for a given source text. Existing methods based on beam search are slow in generating outputs because they insert constraints at separate timesteps in the autoregressive decoding process. We proposed a new method (IG) that encodes two constraints alongside the source text and only generates the words between these constraints.

We evaluated IG against GBS (Hokamp and Liu, 2017) on a gap filling task where either a verb, a noun phrase, or a random sequence of words needed to be generated so as to be compatible with the rest of the translation, which was pre-defined by a (simulated) user. While GBS achieved higher BLEU scores with random sequences, IG achieved higher BLEU scores for verbs and noun phrases, and outperformed GBS by an order of magnitude in terms of speed.

7.2 Practical Implications

Our empirical findings have practical implications for designing MT systems and CAT tools.

7.2.1 MT Systems

Document-level Translation

In both Hassan et al.'s (2018) evaluation of Chinese to English MT and our reassessment in Chapter 3, professional translators did not find a difference in quality between machine and professional human translations of isolated sentences, but, in the latter, showed a significant preference for human translations of full documents. This does not

only imply that human–machine parity assessments should be conducted on the document level (Section 7.3.1), but also that MT systems need to consider document-level context for further improvement. Many of the MT errors in outputs of Hassan et al.'s (2018) sentence-level system, such as incoherent translation of named entities or wrong pronouns (Table 3.6), are owed to the fact that a sentence-level system has no means of conditioning translation decisions in one sentence on translation decisions in other sentences of the same document. Methods for training and decoding with neural models that consider the previous sentence (Tiedemann and Scherrer, 2017) or larger sections of a document (Junczys-Dowmunt, 2019) have shown convincing results in research settings, and should be incorporated into commercial systems for fewer inconsistencies between translated sentences.

Generation of Translation Alternatives

Professional translators do not only want to use MT for increased throughput, but also for inspiration: translators surveyed in this work (Section 4.3.2) as well as by Moorkens and O'Brien (2017) would like to see alternative suggestions for particular words or phrases in translations. Some CAT Tools and online MT services offer such functionality based on prefixes, i.e., they can suggest alternative continuations for a given point in a sentence. This is an instance of prefix decoding (Section 6.2.2), where the translation up to the selected point acts as a constraint for the MT system. The words that follow after the word or phrase a translator requests alternatives for (i.e., the suffix), however, cannot be considered with this mechanism: the suggested alternatives are not conditioned on this suffix, and if an alternative is selected, the suffix is regenerated by the MT system. Our method for infix generation, proposed in Section 6.3, allows the retrieval of word or phrase alternatives given both the beginning and the end of a translation. We showed that efficient infix generation is feasible with current MT technology, and believe it should be made accessible to translators who wish to see translation alternatives within a sentence that will not require changing the rest of the sentence.

Domain Adaptation

MT systems produce translations for new texts by imitating the translations they have seen during training, i.e., the training data. If a text to be translated differs significantly from the training data, the system is likely to produce inadequate translations. In such cases, previous work focused on training domain-specific models, typically with TMs in the given domain, to increase post-editing productivity (e.g., Plitt and Masselot, 2010; Federico et al., 2012; Fischer and Läubli, 2020). Even if the training of domain-specific models is automated, it requires considerable computational resources, and new translations added to the TM are only considered after the next (re)training. The methods we reviewed in Chapter 6 and NFR (Bulté and Tezcan, 2019) in particular allow for real-time inclusion of fuzzy matches (and other translations). NFR was recently shown to achieve

comparable or better quality than model fine-tuning (Xu et al., 2020), paving the way for real-time domain adaptation by means of encoding fuzzy matches along with the source sentence.

7.2.2 CAT Tools

Text Presentation

In the pre-experiment survey of our controlled experiment on text presentation (Chapter 5), 85 % of the subjects stated to use CAT tools with sentence segmentation and left–right orientation in their daily work. This is the default – and often the only available – UI configuration in commercial CAT tools. In our post-experiment survey, in contrast, only 15 % preferred this configuration; the majority preferred sentence segmentation with top–bottom orientation. While our experimental tasks did not mirror real-life translation, as further discussed in Section 7.4, top–bottom orientation also enabled significantly higher speed at no loss of accuracy in an experimental task that was focused on interleaved reading and writing (text reproduction). This leads us to conclude that, at least for translation jobs which involve a significant amount of interleaved reading and writing, CAT tools should arrange source and target sentences with top–bottom rather than left–right orientation.

However, the finding that sentence-level translation quality can be en par with human translation in some languages and domains (Chapter 3; Toral, 2020) implies that, going forward, professional human translation will involve less writing and more reading as translators will revise increasingly better MT suggestions. Our experimental revision task was rather close to real-life revision, and for revising suprasentential discourse relations at least, an unsegmented view of the source and target text enabled higher accuracy compared to UIs that used sentence segmentation. The practical implication here is that CAT tools should, at least for revision, allow translators to work with unsegmented text.

Integration of MT

Now that constrained decoding and data augmentation methods allow for real-time integration of pre-defined partial translations into MT outputs, tighter integration is needed between CAT tools and MT systems. In contrast to TMs and TBs which are integral parts of widely used CAT tools, MT suggestions are mostly pulled in from external systems. Traditionally, CAT tools send out source text and receive target text for entire segments. However, if an external MT system can make use of fuzzy matches, terminology, or user-defined constraints, the prerequisite is that CAT tools – which effectively mediate the interaction between translators and MT systems – elicit and transmit this information to the MT system. The missing support for interaction with more advanced MT technology in commercial CAT tools has motivated – or forced researchers to consider – the implementation of research prototypes from scratch (e.g., Green et al., 2014a; Hokamp and Liu,

2015), and hinders experimentation with CAT tools that translators are used to. In light of the recent advances in MT discussed in Chapter 6, however, we assume that commercial CAT tools will no longer be able to afford restricting the data exchange with MT systems to source and target text alone, unless they integrate their own MT models as a core component just like TMs and TBs.

7.3 Methodological Suggestions

Some of the methodology used in this thesis could be useful in other research settings, and we would like to highlight a number of recommendations and concerns.

7.3.1 Assessment of Translation Quality

We believe that evaluating MT at the document level should become the new normal in benchmarking strong MT systems. Our findings presented in Chapter 3 have motivated a shift towards document-level evaluation in the News Translation task at WMT 2019, a major venue for competitive MT research, where, for some languages, human raters scored translations of full news articles rather than isolated sentences for the first time (Barrault et al., 2019). While this marks the beginning of large-scale document-level evaluation in MT, many details of the evaluation protocol will need refinement.

First, our controlled experiment in Chapter 5 revealed that text presentation has been neglected in software workbenches for professional translators. Similarly, the presentation of news articles at WMT 2019 was arguably suboptimal: raters saw all of the sentences in a news article at once, but without any document structure. In the example shown in Figure 7.1, the first sentence is the title of the news article to be evaluated, but just like paragraph boundaries, this information is lost in the evaluation interface. We believe that document-level evaluation will need to take structural and, to some degree, visual information in documents into account. After all, translation errors may be more severe in a title or an image caption than elsewhere, so raters should be able see such structural information in the documents they evaluate.

Second, document-level evaluation is expensive. Whereas the scoring of 20 sentences in a news article results in 20 data points in a sentence-level evaluation, the scoring of the document as a whole results in a single data point at the same expense, so with a fixed budget, document-level evaluation results in low statistical power compared to sentence-level evaluation (Graham et al., 2019). We would like to point out that the important aspect of document-level evaluation is not the experimental unit, but the availability of visual context. Toral (2020) uses a form of sentence-level scoring where raters are shown the previous and next sentence along the sentence to be scored, but a recent study across three domains and 18 language pairs by Castilho et al. (2020) finds that only 30–60 % of context-related issues can be revealed by showing two preceding sentences, whereas 5–15 % need up to 10 preceding sentences, and ‘10–20 % of issues can be resolved only by

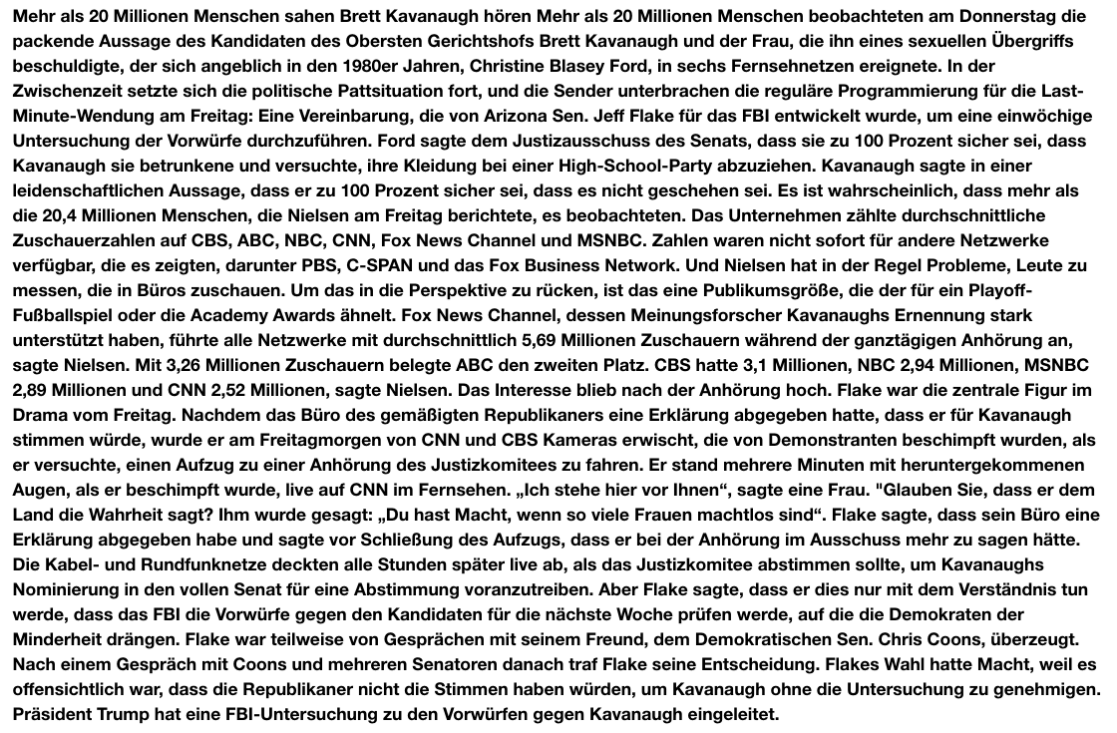


Figure 7.1: German translation of an English news article as shown to raters in the WMT 2019 document-level evaluation campaign (DR+DC). Visual cues such as paragraph boundaries or different font sizes for titles and regular text are missing.

global or visual context'. For future experiments, we encourage the scoring of sentences or paragraphs within a document such that the entire document is visible to the rater at all times.

Third, document-level context is only one of several important aspects in evaluating strong MT systems, particularly if the goal is to assess human–machine parity. In our reassessment of Hassan et al.'s (2018) evaluation, we made two important choices whose impact we did not assess empirically: we employed professional translators instead of bilingual crowd workers as raters, and excluded translationese from the test set. In concurrent work, Toral et al. (2018) verified empirically that these choices have a significant impact on quality scores. Läubli et al. (2020a) combine the findings of Toral et al. (2018) and those presented in Chapter 3 with further experiments and – in addition to using full documents, professional translators, and no translationese – recommend to evaluate fluency in addition to adequacy, and to avoid using human reference translations that are heavily edited for fluency.

7.3.2 Evaluation of User Interfaces

A challenge inherent to assessing the impact of changes to translation technology on translators is that such assessments are bound to a UI, and since fundamental changes to UIs are not possible in commercial CAT tools, the use of purpose-built prototypes is often inevitable. This introduces a confounding variable in experiments: the outcome will not only depend on the feature that is being investigated, but likely also on other aspects in which the prototype differs from CAT tools that translators are familiar with. MT research has a long history of underestimating these factors, as evidenced, for example, in the TransType project. While testing the impact of displaying real-time word completions during translation, Langlais et al. (2000) realised 'that this concept of real-time suggestions depends very much on the usability of the prototype; we had first developed a much simpler editor but its limitations were such that the translators found it unusable.'

With this in mind, we took great care in designing the prototypes for our assessment of how changes in text presentation impact translator performance (Chapter 5). However, even a pilot run with two professional translators did not reveal that we underestimated the impact of the scrolling behaviour in our document-level UIs. As described in more detail in Section 5.4.3, the source and target documents had separate scroll bars, and the fact that scrolling in one document did not invoke automatic scrolling to the relevant position in the other document resulted in a 'a scrolling and matching nightmare' according to one subject. While we considered implementing a real-time alignment mechanism that would enable synchronised scrolling, we were concerned that alignment errors would irritate subjects. With hindsight, we should have compared automatic and manual scrolling in a separate pilot experiment, and choose the one found to be less irritating overall in our main experiment.

Nevertheless, we encourage research where MT is evaluated within UIs and with actual

rather than simulated users, as we did with IG in Section 6.3.2. After all, the distinction of UIs and MT systems 'is artificial in practice since the [two] must work in concert' in real-life translation (Green et al., 2014b). Ideally, future work could establish a collaboration with providers of commercial CAT tools to avoid large differences between prototypes and 'real' software, thus maximising the external validity of experiments with novel translation technology.

7.4 Limitations and Future Work

The experiments presented in this thesis were based on texts from a single domain (news articles) and a single language direction: English to Chinese in Chapter 3, and English to German in Chapters 5 and 6. For human–machine parity assessments in particular, further research will be needed to verify that our findings can be generalised to other languages and domains.

We involved professional translators in all of our investigations, except for the evaluation of IG in Section 6.3.2, where we simulated user input and used automatic metrics for quality evaluation. Future work should close this gap, and involve a large number of subjects where possible. While our controlled full-day experiment involved 20 professional translators, a number similar to those in the largest controlled translation experiments related to MT we are aware of (Green et al., 2013, 2014a), we only recruited 8 professionals for our reassessment of Hassan et al.'s (2018) study in Chapter 3, which resulted in a rather modest sample size. As discussed in Section 7.3.1, the collection of translation quality ratings for full documents is expensive, and future research should explore alternative evaluation protocols that allow more efficient scoring while giving raters access to the linguistic context needed to make valid judgements, possibly by collecting sentence- or paragraph-level ratings in UIs that show these units embedded in their original document. We piloted this setting in our error categorisation in Section 3.5.

A limitation discussed extensively in Sections 5.4.3 and 7.3.2 is that our assessment on the impact of text presentation on translator performance was based on simplistic CAT tool prototypes and tasks that are related to, but do not mirror, real-life translation. Our aim here was to maximise internal validity: because translator productivity is difficult to define and measure (Krings, 2001; House, 2013; Läubli and Green, 2019), we opted for an experimental design in which we could measure translator performance with minimal ambiguity. While this was a conscious decision because it was clear from the outset that we could not implement our alternative UIs into a widely used CAT tool and thus imitate a realistic working environment, future work should do exactly that, possibly through collaboration with commercial CAT tool providers.

7.5 Concluding Remarks

From a bird's eye view, this thesis has shown two things. First, state-of-the-art MT can do much more than translators may think: it can produce high quality sentence-level translation suggestions that live up to their own standards (Chapter 3), integrate terminology and fuzzy matches on the fly, and be configured to only translate the part(s) of a sentence that they wish to delegate to the machine (Chapter 6). Second, translators are much more open to change than translation technology researchers and developers may think: they embrace opportunities to co-create novel features and technology (Chapter 4), and are ready to break long-standing habits – as evidenced by our UI experiment in which, after just one day's work with research prototypes, subjects noted that a fundamental change in text presentation 'worked like a breeze', or that 'I have never arranged my programs this way and I might have to' (Chapter 5).

Professional translators and MT are ready for each other. The real bottleneck, in our opinion, is a lack of research and development on UIs – including those in commercial CAT tools – which mediate the interaction between the two.

Bibliography

Alabau, Vicent, Michael Carl, Mercedes García Martínez, Jesús González-Rubio, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Sofia Rodrigues, and Moritz Schaeffer. 2015. Analysis of the third field trial. Technical Report D6.3, ICT Project 287576 (CASMACAT) of the European Community's Seventh Framework Programme for Research and Technological Development.

Alben, Lauralee. 1996. Quality of experience: Defining the criteria for effective interaction design. *Interactions* 3(3):11–15.

Arif, Ahmed Sabbir and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *Proceedings of TIC-STH*. Toronto, Canada, pages 100–105.

Assis Rosa, Alexandra. 2012. Translating place: Linguistic variation in translation. *Word and Text* 2(2):75–97.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint* 1607.06450.

Baayen, R. Harald and Petar Milin. 2010. Analyzing reaction times. *International Journal of Psychological Research* 3(2):12–28.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*. San Diego, CA, USA.

Banerjee, Satanjeev and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, MI, USA, pages 65–72.

Bar-Hillel, Yehoshua. 1951. The present state of research on mechanical translation. *American Documentation* 2(4):229–237.

Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical Approaches to Computer-assisted Translation. *Computational Linguistics* 35(1):3–28.

Barrault, Loïc, Ond ej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of WMT*. Association for Computational Linguistics, Florence, Italy, pages 128–188.

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67(1):1–48.

Belam, Judith and Derek Lewis. 2002. Report on the 6th EAMT Workshop Teaching Machine Translation. *Machine Translation Review* 13.

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Proceedings of NIPS*. Denver, CO, USA, pages 932–938.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

Bengio, Yoshua, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In Dawn E. Holmes and Lakhmi C. Jain, editors, *Innovations in Machine Learning: Theory and Applications*, Springer, pages 137–186.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: A case study. In *Proceedings of EMNLP*. Austin, TX, USA, pages 257–267.

Bisbey, Richard L. and Martin Kay. 1972. The MIND translation system: A study in man-machine collaboration. Technical Report P-4786, RAND Corporation, Santa Monica, CA, USA.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of WMT*. Sofia, Bulgaria, pages 1–44.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of WMT*. Copenhagen, Denmark, pages 169–214.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of WMT*. Berlin, Germany, pages 131–198.

Bojar, Ondřej, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of WMT*. Belgium, Brussels, pages 272–307.

Bojar, Ondřej, Christian Federmann, Barry Haddow, Philipp Koehn, and Lucia Specia Matt Post and. 2016b. Ten years of WMT evaluation campaigns: Lessons learnt. In *Workshop on Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*. Portorož, Slovenia, pages 27–36.

Bowker, Lynne. 2005. Productivity vs quality? A pilot study on the impact of translation memory systems. *Localisation Focus* 4(1):13–20.

Braun, Virginia and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3(2):77–101.

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, Robert L. Mercer, and Paul Roossin. 1988. A statistical approach to language translation. In *Proceedings of COLING*. pages 71–76.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics* 19(2):263–311.

Bulté, Bram and Arda Tezcan. 2019. Neural Fuzzy Repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of ACL*. Florence, Italy, pages 1800–1809.

Cadwell, Patrick, Sheila Castilho, Sharon O'Brien, and Linda Mitchell. 2016. Human factors in machine translation and post-editing among institutional translators. *Translation Spaces* 5(2):222–243.

Cadwell, Patrick, Sharon O'Brien, and Carlos S. C. Teixeira. 2018. Resistance and accommodation: Factors for the (non-) adoption of machine translation among professional translators. *Perspectives* 26(3):301–321.

Callison-Burch, Chris. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of EMNLP*. Singapore, pages 286–295.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of WMT*. Prague, Czech Republic, pages 136–158.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of BLEU in machine translation research. In *Proceedings of EACL*. Trento, Italy, pages 249–256.

Campbell, Stuart. 1999. A cognitive approach to source text difficulty in translation. *Target* 11(1):33–63.

Carbonell, Jaime R. 1970. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems* 11(4):190–202.

Carl, Michael. 2010. A computational framework for a cognitive model of human translation processes. In *Proceedings of Translating and the Computer*. London, UK.

Castilho, Sheila, Stephen Doherty, Federico Gaspari, and Joss Moorkens. 2018a. Approaches to human and machine translation quality assessment. In *Translation Quality Assessment: From Principles to Practice*, Springer, volume 1 of *Machine Translation: Technologies and Applications*, pages 9–38.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017a. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics* 108:109–120.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Panayota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. 2017b. A comparative quality evaluation of PBSMT and NMT using professional translators. In *Proceedings of MT Summit*. Nagoya, Japan, pages 116–131.

Castilho, Sheila, Joss Moorkens, Federico Gaspari, Rico Sennrich, Andy Way, and Panayota Georgakopoulou. 2018b. Evaluating MT for massive open online courses. *Machine Translation* 22(3):255–278.

Castilho, Sheila, Maja Popović, and Andy Way. 2020. On context span needed for machine translation evaluation. In *Proceedings of LREC*. Marseille, France, pages 3735–3742.

Chen, Stanley F. and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, USA.

Cheng, Shanbo, Shujian Huang, Huadong Chen, Xin-Yu Dai, and Jiajun Chen. 2016. PRIMT: A pick-revise framework for interactive machine translation. In *Proceedings of NAACL*. San Diego, CA, USA, pages 1240–1249.

Cho, Kyunghyun, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar, pages 103–111.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*. Doha, Qatar, pages 1724–1734.

Church, Kenneth W. and Eduard H. Hovy. 1993. Good applications for crummy machine translation. *Machine Translation* 8(4):239–258.

Church, Kenneth Ward and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1):22–29.

Cooper, Alan. 2004. *The Inmates are Running the Asylum: Why Hi-Tech Products Drive Us Crazy and How to Restore*. Sams, Indianapolis, IN, USA.

Coppers, Sven, Jan Van den Bergh, Kris Luyten, Karin Coninx, Iulianna van der Lek-Ciudin, Tom Vanallemeersch, and Vincent Vandeghinste. 2018. Intellingo: An intelligible translation environment. In *Proceedings of CHI*. New York, NY, USA, pages 524:1–524:13.

Crego, Josep, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *arXiv preprint* 1610.05540.

Daems, Joke. 2016. *A translation robot for each translator? A comparative study of manual translation and post-editing of machine translations: Process, quality and translator attitude*. Ph.D. thesis, Ghent University, Ghent, Belgium.

Daems, Joke, Orphée De Clercq, and Lieve Macken. 2017. Translationese and post-editese: How comparable is comparable quality? *Linguistica Antverpiensia, New Series – Themes in Translation Studies* 16:89–103.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1):1–22.

Ding, Shuoyang, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. 2016. The JHU machine translation systems for WMT 2016. In *Proceedings of WMT*. Berlin, Germany, pages 272–280.

Dinu, Georgiana, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of ACL*. Florence, Italy, pages 3063–3068.

do Carmo, Félix and Joss Moorkens. 2020. Differentiating editing, post-editing and revision. In Maarit Koponen, Brian Mossop, Isabelle S. Robert, and Giovanna Scocchera, editors, *Translation Revision and Post-editing: Industry practices and cognitive processes*, Routledge.

Dragsted, Barbara. 2006. Computer-aided translation as a distributed cognitive task. *Pragmatics & Cognition* 14(2):443–464.

Dragsted, Barbara. 2010. Coordination of reading and writing processes in translation.

In Gregory M. Shreve and Erik Angelone, editors, *Translation and Cognition*, John Benjamins, pages 41–62.

Durrani, Nadir, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of WMT*. Baltimore, MD, USA, pages 97–104.

Ehrensberger-Dow, Maureen, Andrea Hunziker Heeb, Gary Massey, Ursula Meidert, Silke Neumann, and Heidrun Becker. 2016. An international survey of the ergonomics of professional translation. *Revue de l'Institut des langues et cultures d'Europe, Amérique, Afrique, Asie et Australie (ILCEA)* 27.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science* 14(2):179–211.

Emerson, John D. and Gary A. Simon. 1979. Another look at the sign test when ties are present: The problem of confidence intervals. *The American Statistician* 33(3):140–142.

Engelbart, Douglas C. and William K. English. 1968. A research center for augmenting human intellect. In *Proceedings of AFIPS*. San Francisco, CA, USA, pages 395–410.

Ericsson, K. Anders and Herbert A. Simon. 1984. *Protocol Analysis: Verbal Reports as Data*. MIT Press.

Esteban, José, José Lorenzo, Antonio S. Valderrábanos, and Guy Lapalme. 2004. TransType2 – an innovative computer-assisted translation system. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics (ACL)*. Barcelona, Spain, pages 94–97.

Etchegoyhen, Thierry, Lindsay Bywood, Mark Fishel, Panayota Georgakopoulou, Jie Jiang, Gerard Van Loenhout, Arantza Del Pozo, Mirjam Sepesy Maucec, Anja Turner, and Martin Volk. 2014. Machine translation for subtitling: A large-scale evaluation. In *Proceedings of LREC*. Reykjavik, Iceland, pages 46–53.

Federico, Marcello, Alessandro Cattelan, and Marco Trombetti. 2012. Measuring user productivity in machine translation enhanced computer assisted translation. In *Proceedings of AMTA*. San Diego, CA, USA.

Firat, Orhan, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of EMNLP*. Austin, TX, USA, pages 268–277.

Fischer, Lukas and Samuel Läubli. 2020. What's the difference between professional human and machine translation? A blind multi-language study on domain-specific MT. In *Proceedings of EAMT*. Lisbon, Portugal, pages 215–224.

Flanagan, Kevin. 2014. *Methods for improvising subsegment recall in translation memory*. Ph.D. thesis, Swansea University, Swansea, UK.

Flournoy, Raymond and Christine Duran. 2009. Machine translation and document loc-

alization at adobe: From pilot to production. *Proceedings of MT Summit* pages 425–428.

Fomicheva, Marina and Lucia Specia. 2016. Reference bias in monolingual machine translation evaluation. In *Proceedings of ACL*. Berlin, Germany, pages 77–82.

Foster, George, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation* 12(1):175–194.

Francis, Wendy S. and Silvia P. Sáenz. 2007. Repetition priming endurance in picture naming and translation: Contributions of component processes. *Memory & Cognition* 35(3):481–493.

Freitag, Markus, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. *arXiv preprint* 2004.06063.

Gaspari, Federico, Antonio Toral, Sudip Kumar Naskar, Declan Groves, and Andy Way. 2014. Perception vs reality: Measuring machine translation post-editing productivity. In *Proceedings of WPTP*. pages 60–72.

Gelman, Andrew and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Gouadec, Daniel. 2010. Quality in translation. In Yves Gambier and Luc van Doorslaer, editors, *Handbook of Translation Studies*, John Benjamins, volume 1, pages 270–275.

Graham, Yvette, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*. Sofia, Bulgaria, pages 33–41.

Graham, Yvette, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone? *Natural Language Engineering* 23(1):3–30.

Graham, Yvette, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *arXiv preprint* 1906.09833.

Green, Spence, Jason Chuang, Jeffrey Heer, and Christopher D. Manning. 2014a. Predictive Translation Memory: A mixed-initiative system for human language translation. In *Proceedings of UIST*. Honolulu, Hawaii, USA, pages 177–187.

Green, Spence, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of CHI*. Paris, France.

Green, Spence, Sida Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014b. Human effort and machine learnability in computer aided translation. In *Proceedings of EMNLP*. Doha, Qatar, pages 1225–1236.

Gu, Jiatao, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Proceedings of ICLR*. Vancouver, Canada.

Gu, Jiatao, Changhan Wang, and Junbo Zhao. 2019. Levenshtein Transformer. In *Proceedings of NIPS*. Vancouver, Canada.

Haque, Rejwanul, Md Hasanuzzaman, and Andy Way. 2019. Investigating terminology translation in statistical and neural machine translation: A case study on English-to-Hindi and Hindi-to-English. In *Proceedings of RANLP*. Varna, Bulgaria, pages 437–446.

Hassan, Hany, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint* 1803.05567.

Heer, Jeffrey. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116(6):1844–1850.

Hieber, Felix, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of EAMT*. Lisbon, Portugal, pages 457–458.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Hokamp, Chris and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of ACL*. Vancouver, Canada, pages 1535–1546.

Hokamp, Christopher and Qun Liu. 2015. HandyCAT - an open-source platform for CAT tool research. In *Proceedings of EAMT*. Antalya, Turkey, page 216.

Hornbæk, Kasper and Erik Frøkjær. 2001. Reading of electronic documents: The usability of linear, fisheye, and overview+detail interfaces. In *Proceedings of CHI*. Seattle, WA, USA, pages 293–300.

House, Juliane. 2013. Quality in translation studies. In Carmen Millán and Francesca Bartrina, editors, *The Routledge Handbook of Translation*, Routledge, chapter 39, pages 534–547.

Hutchins, John. 1997. From first conception to first demonstration: The nascent years of machine translation, 1947–1954. A chronology. *Machine Translation* 12(3):195–252.

ISO 17100:2015. 2015. Translation services – Requirements for translation services. Standard, International Organization for Standardization, Geneva, Switzerland.

ISO 18587:2017. 2017. Translation services – Post-editing of machine translation

output – Requirements. Standard, International Organization for Standardization, Geneva, Switzerland.

ISO 9241-210:2010. 2017. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Standard, International Organization for Standardization, Geneva, Switzerland.

Jakobsen, Arnt Lykke. 2003. Effects of think aloud on translation speed, revision and segmentation. In Fabio Alves, editor, *Triangulating Translation: Perspectives in process oriented research*, John Benjamins, number 45 in Benjamins Translation Library, pages 69–95.

Junczys-Dowmunt, Marcin. 2019. Microsoft Translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of WMT*. Florence, Italy, pages 225–233.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL*. Melbourne, Australia, pages 116–121.

Karimova, Sariya, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation* 32:309–324.

Kay, Martin. 1980. The proper place of men and machines in language translation. Research report CSL-80-11, Xerox Palo Alto Research Center, Palo Alto, CA, USA.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*. San Diego, CA, USA.

Kittur, Aniket, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of CHI*. Florence, Italy, pages 453–456.

Kneser, Reinhard and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *Proceedings of ICASSP*. volume 1, pages 181–184.

Knowles, Rebecca and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of AMTA*. Austin, TX, USA, pages 107–120.

Koehn, Philipp. 2009. A process study of computer-aided translation. *Machine Translation* 23(4):241–263.

Koehn, Philipp. 2010. *Statistical Machine Translation*. Cambridge University Press.

Koehn, Philipp, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of IWSLT*. Pittsburgh, PA, USA.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Proceedings of EMNLP/CoNLL*. Prague, Czech Republic, pages 868–876.

Koehn, Philipp and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings of WMT*. New York, NY, USA, pages 102–121.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*. Edmonton, Canada, pages 48–54.

Koehn, Philipp and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of the 2nd joint EM+/CNGL Workshop "Bringing MT to the user: Research on integrating MT in the translation industry"*. Denver, Colorado, USA, pages 21–31.

Krings, Hans P. 1994. *Texte reparieren: Empirische Untersuchungen zum Prozeß der Nachredaktion von Maschinenübersetzungen*. Habilitation thesis, Universität Hildesheim, Hildesheim, Germany.

Krings, Hans P. 2001. *Repairing Texts: Empirical Investigations of Machine Translationg Post-Editing Processes*. Kent State University Press, Kent, OH, USA.

Kudo, Taku and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP*. Brussels, Belgium, pages 66–71.

Kurokawa, David, Cyril Goutte, and Pierre Isabelle. 2009. Automatic detection of translated text and its impact on machine translation. In *Proceedings of MT Summit*. Ottawa, Canada, pages 81–88.

Langlais, Philippe, George Foster, and Guy Lapalme. 2000. TransType: A computer-aided translation typing system. In *Proceedings of the 2000 ANLP-NAACL Workshop on Embedded Machine Translation Systems*. Seattle, WA, USA, pages 46–51.

Langlais, Philippe and Guy Lapalme. 2002. TransType: Development-evaluation cycles to boost translator's productivity. *Machine Translation* 17(2):77–98.

Langlais, Philippe, Guy Lapalme, and Sébastien Sauvé. 2001. User interface aspects of a translation typing system. In Eleni Stroulia and Stan Matwin, editors, *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*. Halifax, Canada, pages 246–256.

Laviosa, Sara. 1998. Core Patterns of Lexical Use in a Comparable Corpus of English Narrative Prose. *Meta* 43(4):557–570.

LeBlanc, Matthieu. 2013. Translators on translation memory (TM). Results of an ethnographic study in three translation services and agencies. *Translation & Interpreting* 5(2):1–13.

Lee, Jason, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-

autoregressive neural sequence modeling by iterative refinement. In *Proceedings of EMNLP*. Brussels, Belgium, pages 1173–1182.

Levin, Pavel, Nishikant Dhanuka, and Maxim Khalilov. 2017. Machine translation at booking.com: Journey and lessons learned. In *Proceedings of EAMT*. Prague, Czech Republic, pages 80–85.

Licklider, Joseph Carl Robnett. 1960. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics* HFE-1:4–11.

LISA QA. 2006. LISA QA Model 3.1. Standard, Localization Industry Standards Association, Carouge, Switzerland.

Luong, Minh-Thang and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of IWSLT*. Da Nang, Vietnam.

Läubli, Samuel, Chantal Amrhein, Patrick Düggelin, Beatriz Gonzalez, Alena Zwahlen, and Martin Volk. 2019. Post-editing productivity with neural machine translation: An empirical assessment of speed and quality in the banking and finance domain. In *Proceedings of Machine Translation Summit XVII*. Dublin, Ireland, pages 267–272.

Läubli, Samuel, Sheila Casthilo, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020a. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research* 67:653–672.

Läubli, Samuel, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk. 2013. Assessing post-editing efficiency in a realistic translation environment. In *Proceedings of WPTP*. Nice, France, pages 83–91.

Läubli, Samuel and Spence Green. 2019. Translation technology research and human–computer interaction. In Minako O'Hagan, editor, *The Routledge Handbook of Translation and Technology*, Routledge, chapter 22, pages 370–383.

Läubli, Samuel, Matthias Müller, Beat Horat, and Martin Volk. 2018a. mtrain: A convenience tool for machine translation. In *Proceedings of EAMT*. Alacant, Spain, page 357.

Läubli, Samuel and David Orrego-Carmona. 2017. When Google Translate is better than some human colleagues, those people are no longer colleagues. In *Proceedings of Translating and the Computer*. London, UK, pages 59–69.

Läubli, Samuel, Rico Sennrich, and Martin Volk. 2018b. Has machine translation achieved human parity? A case for document-level evaluation. In *Proceedings of EMNLP*. Brussels, Belgium, pages 4791–4796.

Läubli, Samuel, Patrick Simianer, Joern Wuebker, Geza Kovacs, Rico Sennrich, and Spence Green. 2020b. The impact of text presentation on translator performance. Under review.

Ma, Qingsong, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the

WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of WMT*. Florence, Italy, pages 62–90.

Macklovitch, Elliott. 2006. TransType2: The last word. In *Proceedings of LREC*. Genoa, Italy, pages 167–172.

Macklovitch, Elliott, Ngoc Tran Nguyen, and Guy Lapalme. 2005. Tracing translations in the making. In *Proceedings of MT Summit*. Phuket, Thailand, pages 323–330.

Mikolov, Tomáš, Martin Karafiát, Lukáš Burget, Jan Černock , and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*.

Mikolov, Tomáš, Stefan Kombrink, Lukáš Burget, Jan Černocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of ICASSP*. Prague, Czech Republic, pages 5528–5531.

Miniukovich, Aliaksei, Antonella De Angeli, Simone Sulpizio, and Paola Venuti. 2017. Design guidelines for web readability. In *Proceedings of DIS*. Edinburgh, UK, pages 285–296.

Moore, William L. 1982. Concept testing. *Journal of Business Research* 10(3):279–294.

Moorkens, Joss and Sharon O'Brien. 2017. Assessing user interface needs of post-editors of machine translation. In Dorothy Kenny, editor, *Human Issues in Translation Technology*, Routledge, pages 109–130.

Moorkens, Joss, Antonio Toral, Sheila Castilho, and Andy Way. 2018. Translators' perceptions of literary post-editing using statistical and neural machine translation. *Translation Spaces* 7(2):240–262.

MQM. 2015. Multidimensional Quality Metrics. Standard, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Saarbrücken, Germany.

Müller, Mathias, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of WMT*. Brussels, Belgium, pages 61–72.

Neubig, Graham, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of WAT2015*. Kyoto, Japan.

Ng, Nathan, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of WMT*. Florence, Italy, pages 314–319.

Norman, Don. 1988. *The Design of Everyday Things*. Basic Books.

O'Brien, Sharon. 2009. Eye tracking in translation process research: Methodological challenges and solutions. In Inger M. Mees, Fabio Alves, and Susanne Göpferich, editors, *Methodology, technology and innovation in translation process research*, Samfundslitteratur, volume 38, pages 251–266.

O'Brien, Sharon. 2012. Translation as human–computer interaction. *Translation Spaces* 1(1):101–122.

O'Brien, Sharon, Maureen Ehrensberger-Dow, Marcel Hasler, and Megan Connolly. 2017. Irritating CAT tool features that matter to translators. *HERMES* 56:145–162.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*. Sapporo, Japan, pages 160–167.

Olohan, Maeve. 2007. Economic trends and developments in the translation industry. *The Interpreter and Translator Trainer* 1(1):37–63.

Orrego-Carmona, David. 2016. A reception study on non-professional subtitling: Do audiences notice any difference? *Across Languages and Cultures* 17(2):163–181.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*. Philadelphia, PA, USA, pages 311–318.

Parra Escartín, Carla and Manuel Arcedillo. 2015. Machine translation evaluation made fuzzier: A study on post-editing productivity and evaluation metrics in commercial settings. In *Proceedings of MT Summit*. Miami, FL, USA, pages 131–144.

Peris, Álvaro, Luis Cebrián, and Francisco Casacuberta. 2017. Online learning for neural machine translation post-editing. *arXiv preprint* 1706.03196.

Pielmeier, Hélène and Arle Lommel. 2019. Machine translation use at LSPs: Data on how language service providers use MT. Research report, CSA Research, Cambridge, MA, USA.

Pierce, John R., John B. Carroll, Eric P. Hamp, David G. Hays, Charles F. Hockett, Anthony G. Oettinger, and Alan Perlis. 1966. Language and machines: Computers in translation and linguistics. Research report, Automatic Language Processing Advisory Committee (ALPAC), Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, D.C.

Plitt, Mirko and François Masselot. 2010. A productivity test of statistical machine translation post-editing in a typical localisation context. *Prague Bulletin of Mathematical Linguistics* 93:7–16.

Plumb, Robert K. 1954. Russian is turned into English by a fast electronic translator. New York Times, 8 January 1956, pages 1; 5.

Popel, Martin, Dominik Macháček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. 2019. English–Czech systems in WMT19: Document-level Transformer. In *Proceedings of WMT*. Florence, Italy, pages 342–348.

Post, Matt and David Vilar. 2018. Fast lexically constrained decoding with Dynamic Beam Allocation for neural machine translation. In *Proceedings of NAACL*. New Orleans, LA, USA, pages 1314–1324.

Rello, Luz, Martin Pielot, and Mari-Carmen Marcos. 2016. Make it big!: The effect of font size and line spacing on online readability. In *Proceedings of CHI*. San Jose, CA, USA, pages 3637–3648.

Roberts, Teresa L. 1980. *Evaluation of Computer Text Editors*. Ph.D. thesis, Stanford University, Stanford, CA, USA.

Roberts, Teresa L. and Thomas P. Moran. 1982. Evaluation of text editors. In *Proceedings of CHI*. Gaithersburg, MD, USA, pages 136–141.

Ruiz, Carmen, Natalia Paredes, Pedro Macizo, and Maria Teresa Bajo. 2008. Activation of lexical and syntactic target language properties in translation. *Acta Psychologica* 128(3):490 – 500.

SAE J2450. 2005. Quality Metric for Language Translation of Service Information. Standard, SAE International, Warrendale, PA, USA.

Saffer, Dan. 2005. *The Role of Metaphor in Interaction Design*. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

Sager, Juan C. 1994. *Language Engineering and Translation: Consequences of automation*. John Benjamins.

Sakaguchi, Keisuke, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of WMT*. Baltimore, MD, USA, pages 1–11.

Schneider, Dominik, Marcos Zampieri, and Josef van Genabith. 2018. Translation memories and the translator: A report on a user survey. *Babel* 64(5/6):734–762.

Sennrich, Rico. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of EACL*. Valencia, Spain, pages 376–382.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: A toolkit for neural machine translation. In *Proceedings of EACL*. Valencia, Spain, pages 65–68.

Sennrich, Rico and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of WMT*. Berlin, Germany, pages 83–91.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of WMT*. Berlin, Germany, pages 371–376.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of ACL*. Berlin, Germany, pages 1715–1725.

Shih, Claire Yi-yi. 2006. Revision from translators' point of view: An interview study. *Target* 18(2):295–312.

Shneiderman, Ben. 1983. Direct manipulation: A step beyond programming languages. *Computer* 8:57–69.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*. Cambridge, MA, USA.

Somers, Harold. 2003. Translation memory systems. In Harold Somers, editor, *Computers and Translation: A translator's guide*, John Benjamins, volume 3 of *Benjamins Translation Library*, chapter 3, pages 31–47.

Steinberger, Ralf, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of LREC*. Istanbul, Turkey, pages 454–459.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proceedings of Interspeech*. Portland, OR, USA, pages 194–197.

Susanto, Raymond Hendy, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein Transformer. In *Proceedings of ACL*. pages 3536–3543.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*. Montreal, Canada, pages 3104–3112.

Sánchez-Gijón, Pilar, Joss Moorkens, and Andy Way. 2019. Post-editing neural machine translation versus translation memory segments. *Machine Translation* 33:31–59.

Tiedemann, Jörg and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*. Copenhagen, Denmark, pages 82–92.

Toral, Antonio. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. In *Proceedings of EAMT*. Lisbon, Portugal, pages 185–194.

Toral, Antonio, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of WMT*. Brussels, Belgium, pages 113–123.

Tu, Zhaopeng, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL*. Berlin, Germany, pages 76–85.

Vardaro, Jennifer, Moritz Schaeffer, and Silvia Hansen-Schirra. 2019. Comparing the quality of neural machine translation and professional post-editing. In *Proceedings of QoMEX*. Berlin, Germany.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*. Long Beach, CA, USA, pages 5998–6008.

Vilar, David, Jia Xu, D'Haro Luis Fernando, and Hermann Ney. 2006. Error analysis of statistical machine translation output. In *Proceedings of LREC*. Genoa, Italy, pages 697–702.

Wang, Yuguang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for WMT17. In *Proceedings of WMT*. Copenhagen, Denmark, pages 410–415.

Weaver, Warren. 1947. Letter to Norbert Wiener, 4 march 1947. Reproduced by permission of the Rockefeller Foundation Archives, available at `http://www.mt-archive.info/Weaver-1947-original.pdf`.

Wichmann, Søren, Eric W. Holman, Dik Bakker, and Cecil H. Brown. 2010. Evaluating linguistic distance measures. *Physica A: Statistical Mechanics and its Applications* 389(17):3632–3639.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint* 1609.08144v2.

Wuebker, Joern, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of EMNLP*. Brussels, Belgium, pages 881–886.

Xu, Jitao, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of ACL*. Online, pages 1580–1590.

Yamada, Masaru. 2019. The impact of google neural machine translation on post-editing by student translators. *The Journal of Specialised Translation* 31:87–106.

Yu, Chen-Hsiang and Robert C Miller. 2010. Enhancing web page readability for non-native readers. In *Proceedings of CHI*. Atlanta, GA, USA, pages 2523–2532.

Zaretskaya, Anna. 2015. User requirement analysis. Research report D2.1, EXPERT Project, the European Union's Seventh Framework Programme (FP7).

Zoph, Barret and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of NAACL*. San Diego, CA, USA, pages 30–34.

Appendix A

Whiteboards

Collaborative sketching of interviewer (black ink) and participants (colored).

Figure A.1: Collaborative sketching with P1.

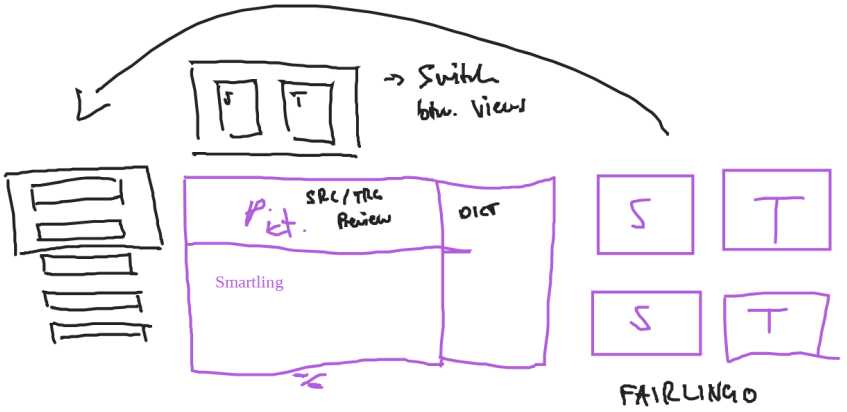


Figure A.2: Collaborative sketching with P2.

Figure A.3: Collaborative sketching with P3.

Figure A.4: Collaborative sketching with P4.

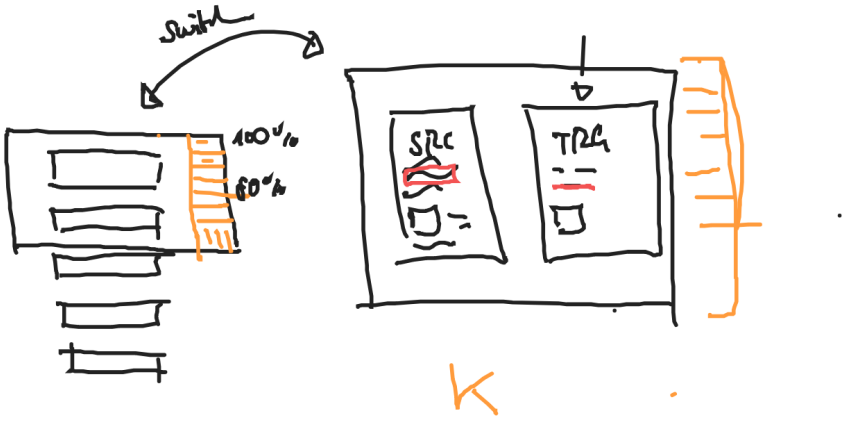


Figure A.5: Collaborative sketching with P5.

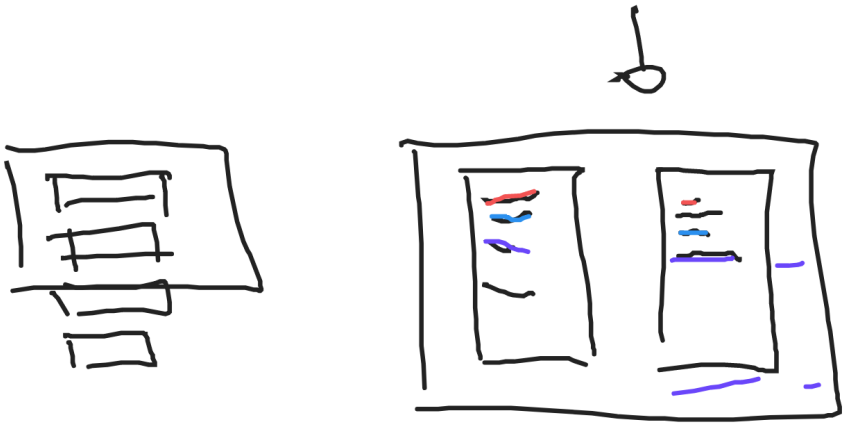


Figure A.6: Collaborative sketching with P6.

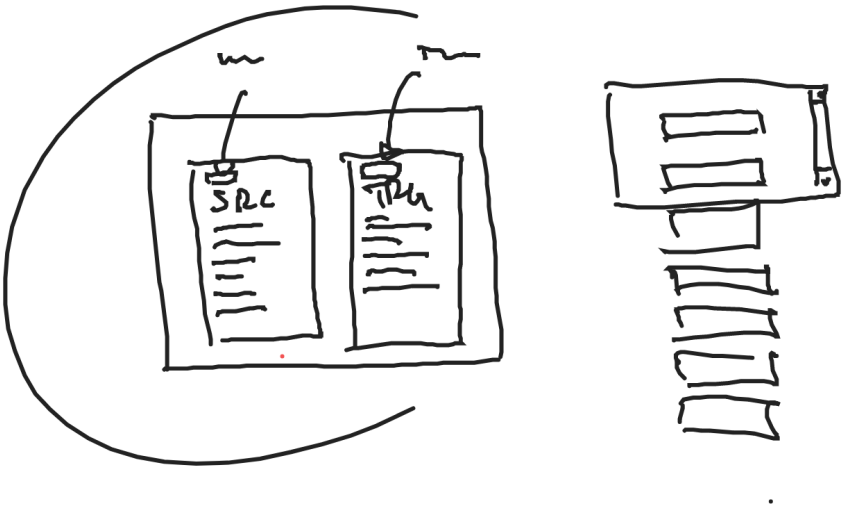


Figure A.7: Collaborative sketching with P7.

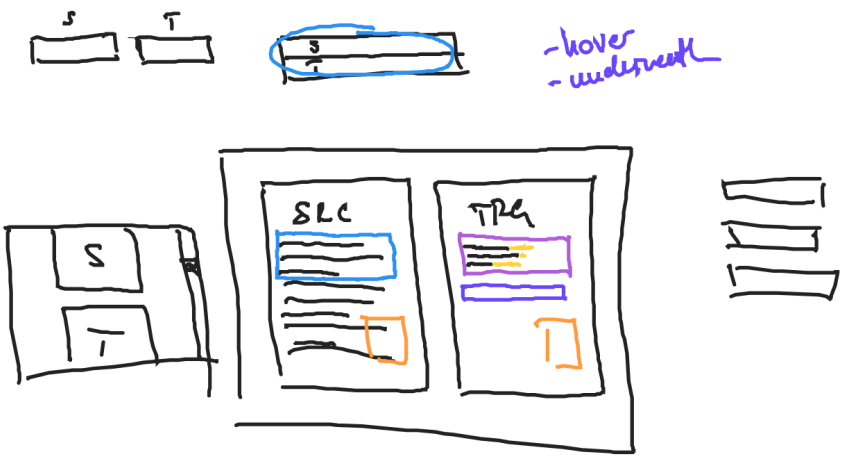


Figure A.8: (= Figure 4.3) Collaborative sketching with P8.